Test Suite for the
CAx Interoperability Forum
Round 57J

September 2025 – March 2026

Version 1.2

17 February 2026

Contacts

Jochen Boy	Asa Trainer
PROSTEP AG	Trainer Engineering Associates
jochen.boy@prostep.com	agtrainer@comcast.net

Table of Contents

List of Figures

Document History

Version	Date	Change
0.1	2025-12-03	Initial Draft
1.0	2025-12-18	Public Release
1.1	2026-01-14	Update statistics definitions for Persistent ID test cases
1.2	2026-02-17	Updated native models for PDU test case

1 Introduction

The CAx Interoperability Forum (CAx-IF) is part of the Model-Based Interoperability Forum (MBx-IF), which is a joint effort between AFNeT, PDES, Inc. and prostep ivip. An interoperability forum is a logical collection of a user group and an implementer group, focused on specific capabilities of a named standard, in this case ISO 10303 STEP.

- The **User Group** is comprised of industry representatives, all members of at least one of the Interoperability Forum hosting organizations. The group will define and prioritize use cases, derive requirements and related validation properties as well as document user best practices.
- The **Implementor Group** is a group of software vendors, 3rd party integrators, and independent implementors, all members of at least one of the Interoperability Forum hosting organizations, that define recommended practices based on the prioritized use cases provided by the user group and validate them in test rounds.

The objectives of the CAx-IF concentrate primarily on testing the interoperability and compliance of STEP processors based on all Editions of AP242, and include documenting and prioritizing use cases, requirements and best practices to ensure completeness and consistency of the STEP standard and it implementations, implementing new functionalities based on users' requirements while ensuring these do not adversely affect existing implementations, avoiding roadblocks by establishing agreed-upon approaches, and increasing user confidence in STEP by providing interoperable commercial software products.

The CAx-IF's Implementor Group performs two test rounds per year for each domain and presents summary results to the user community. Furthermore, Recommended Practices are developed, and issues are reported to the standards development community.

The test rounds in general combine testing of synthetic and production models. Production models will in most cases be provided by the user companies of the organizations AFNeT, PDES, Inc., and prostep ivip Association. When production models are not available from the user companies, "production-like" models will be solicited from the various CAx-IF participants.

This test suite includes synthetic models for testing the following capabilities: Product Manufacturing Information (PMI), both as Graphic Presentation and as Semantic Representation, 3D Tessellated Geometry, Kinematics, Composite Materials, Assembly Structure with External References as well as Kinematic Mechanism definitions in AP242 Domain Model XML format, and Persistent Entity IDs.

1.1 Functionality tested in this Test Round

Functionality tested in this round relates to:

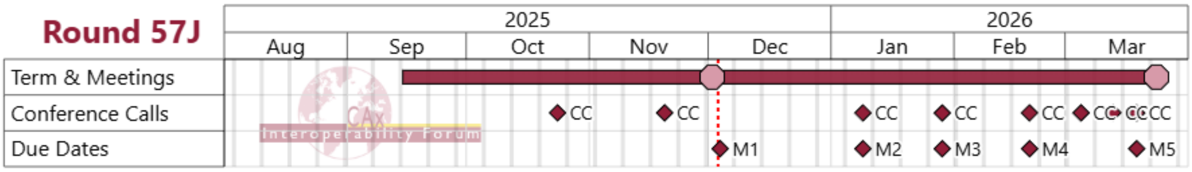
- **Product Manufacturing Information (PMI)** describes the capability to embed information about dimensions, tolerances and other parameters which are necessary input for the manufacturing and measuring of the part from the 3D model. In this round, the focus will be on the two approaches for the transfer of PMI in the 3D model:
    - "Tessellated Presentation" refers to breaking down each annotation into tessellated elements as supported by AP242 and exchanging them as geometry. This preserves the exact shape of the annotation but is human readable only.
    - "Semantic Representation" refers to the intelligent transfer of PMI data in an associative and reusable way. This scenario aims towards driving downstream usage and later modifications of the model. The data is machine-readable, but not necessarily visible in the 3D model. The test also includes additional presentation data, which can be linked to the corresponding PMI representation.
    - "Assembly-level PMI" applies the concept above to assemblies, where PMI elements are defined between different parts, or part instances. Concepts such as Saved Views and Cross-Highlighting shall work in the same way.
- **AP242 Domain Model XML** is an implementation format introduced with AP242, and the designated process format for many applications in the aerospace and automotive industries. It will be used in combination with geometry formats matching the respective requirement. In the CAx-IF, the geometry files will be in STEP Part 21 format. The XML files contain the assembly structure and part master information. The tests, which are conducted jointly with the PDM-IF, primarily aim at improving CAx-PDM interoperability by ensuring that the different types of systems correctly cope with the different levels of information.
- **Composite Materials** are made by layering various plies of material on top of each other. They can be defined in an implicit-precise way, by giving the laminate tables, ply boundaries, orientation, materials, and laminated cores; or in an explicit-tessellated way by calculating the resulting 3D Tessellated Solid. Both representations can be linked to each other.
- **Kinematics** is a capability in AP242 that allows describing the motion of parts over time and in relation to each other. This includes the definition of mechanisms with joints and constraints, defining the kinematic relationships between the parts, as well as motions, which are defined by capturing the positions of the moving parts at discrete points in time. To cover Aerospace as well as Automotive use cases, and to increase the range of participating systems, this capability is being tested jointly with the JT-IF.
- **Persistent Entity IDs** enable the ability to track a product's model information during either design iteration or downstream consumption. This will allow consuming CAx applications to update their designs or manufacturing and inspection applications based on external models when changes are received. It also allows traceability of product information within CAx and PLM systems for forensic analyses.

1.2 General testing instructions for this round

The general procedures for communication of models and statistics are outlined in a separate document, entitled 'General Testing Instructions'. The document can be retrieved from the CAx Interoperability Forum web sites. The latest version is v2.0, dated 23 June 2023.

1.3 Testing Schedule

The following schedule has been agreed upon for Round 57J:



M1: Test Suite and Rec. Pracs. available - M2: Initial STEP files and native stats due
M3: Initial release of files for testing - M4: Initial target stats due - M5: Preliminary results available

Date	Action
23 Oct 2025 (Thu)	Round 56J Follow-up / Round 57J Preparation Call
19 Nov 2025 (Wed)	Round 56J Follow-up / Round 57J Preparation Call
3 Dec 2025 (Wed)	CAx-IF Round 57J Kick-Off Meeting / Test Suite and Rec. Pracs. available
8 Jan 2026 (Thu)	1st CAx-IF Round 57J Conference Call / Initial STEP files and native stats due
28 Jan 2026 (Wed)	2nd CAx-IF Round 57J Conference Call / Initial release of files for testing
19 Feb 2026 (Thu)	3rd CAx-IF Round 57J Conference Call / Initial target stats due
4 Mar 2026 (Wed)	4th CAx-IF Round 57J Conference Call
12 Mar 2026 (Thu)	5th CAx-IF Round 57J Conference Call
18 Mar 2026 (Wed)	6th CAx-IF Round 57J Conference Call / Preliminary results available
24 Mar 2026 (Tue) - 26 Mar 2026 (Thu)	CAx-IF Round 57J Review Meeting in Charleston, SC, USA

Figure 1: CAx-IF Round 57J Schedule

1.4 Copyrights on Test Cases

1.4.1 CAx-IF

None of the production test cases which were provided by the AFNeT, PDES, Inc. and prostep ivip member companies may be publicly released for any purpose. The test cases can be freely distributed among the CAx-IF members and can be used for any purposes that are related to CAx-IF testing (i.e., testing, documentation of testing efforts, etc.), if a reference to the originating company is made.

The test cases must not be used for any purposes other than CAx-IF testing or outside of AFNeT, PDES, Inc. and prostep ivip. Test cases provided by the LOTAR project for testing of specific capabilities are applicable to the same restrictions and may not be used outside LOTAR or the CAx-IF.

1.4.2 NIST

The test cases developed at the National Institute of Standards and Technology (NIST) are not subject to copyright protection and are in the public domain. NIST assumes no responsibility for the components of the test system for use by other parties and makes no guarantees, expressed or implied, about their quality, reliability, or any other characteristic. The use of the CAD systems to create the Test Models does not imply a recommendation or endorsement by NIST.

For more details, read the disclaimer at https://go.usa.gov/xuh9n

1.4.3 JAMA

All copyrights for the sample data and the accompanying explanatory materials are owned by the Japan Automobile Manufacturers Association (JAMA). The data may not be published, distributed, or sold, whether for profit or non-profit, without JAMA's permission, particularly if only minor changes that do not demonstrate significant creativity have been made. However, note that JAMA has approved the use of this sample data for CAx-IF testing.

JAMA does not take any responsibility or liability for any direct or indirect damages incurred by users or third parties resulting from the use of the published data. Users who download the data must comply with these terms as long as they retain the data, regardless of the storage medium (e.g., hard disks, CD-ROMs). The content of the data may be changed or withdrawn without notice.

2 Synthetic Test Case Specifications

2.1 Test Case AP1: Assembly PMI (Domain Model) with EER

All information about this test case can also be viewed in CAESAR on its Information page.

2.1.1 Motivation

The motivation for testing Product and Manufacturing Information (PMI) is the same as described in the JPMI test case, section 2.5.1.

Testing of PMI data exchange has been focused on single parts so far. The next step will be to extend this capability to cover PMI defined at the assembly level. Based on the requirements defined by the user community, all assembly information (the product structure as well as the PMI defined between different components) shall be exchanged as AP242 Domain Model XML files.

Since the Domain Model does not support geometry, the part shapes will still be exchanged as STEP Part 21 files. For a complete definition of the PMI at the assembly level, External Element References (EER) is required from the Domain Model XML assembly file to the Part 21 geometry files. This concept has already been proven in LOTAR pilot projects.

In Round 56J, the goal is to exchange the entire assembly structure in a single AP242 Domain Model XML file.

2.1.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#191	Semantic geometric dimension and tolerance link to assembly occurrence

2.1.3 Approach

The implementation guidelines to define the PMI at the assembly level as well as the EER to the component geometries are provided in the following documents:

- Rec. Practices for AP242 Domain Model XML Product & Assembly Structure
    - v4.0 (Public MBx-IF homepage)
- Rec. Practices for AP242 Domain Model XML Product Manufacturing Information (PMI)
    - v0.30_GHi9
- Presentation “XML Rec. Pracs for PMI”
    - v0.19
- Presentation “EER use cases PDM, PMI and Kinematics”
    - v0.19

All these documents can be found on Nextcloud, folder

MBX-IF > CAX-IF > Draft Recommended Practices

Note that these documents fully support the final published AP242 Edition 4 schemas.

Implementation of the External Element References

- Domain Model XML: For the outgoing references, the approach supported by AP242 Ed.4 (Part 15 Ed.2) shall be used.
- MIM Part 21: For the incoming references, UUIDs in the Data Section shall be used (i.e., no Anchors).

The tests in Round 57J shall use the final AP242 Edition 4 XSD and EXP schemas, which are linked on the MBx-IF homepage under MBx > Resources.

2.1.4 Testing Instructions

Testing of Assembly PMI in Round 56J will be based on prostep ivip’s “Vise” test model. This is an assembly with components and one sub-assembly. PMI is defined at part level as well as at assembly level.

The files are stored on the Nextcloud folder:

MBX-IF > CAX-IF > CAX-IG > Round 55J > AP1

Native CAD files for this test model are provided in three different formats:

- CATIA V5 R28
- Creo 7.04
- NX 2019

These are the same models as last round. In addition to the native CAD files, a PDF document is provided that illustrates all Saved Views for all component parts, as well as the PMI and views defined at the assembly level.

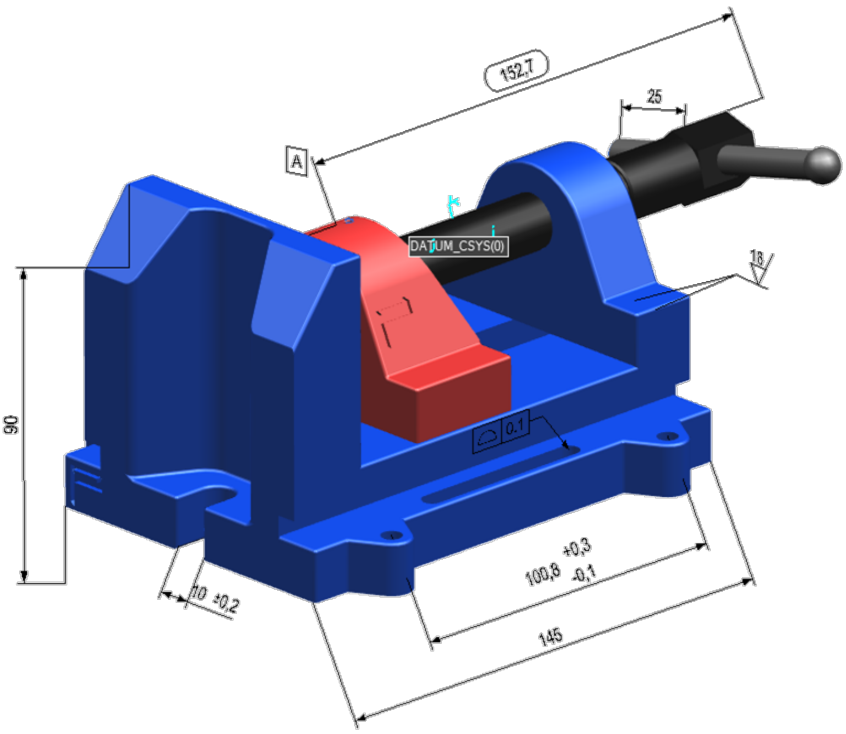


Figure 2: Illustration of the "Vise" test model

2.1.5 Statistics

For each STEP file exported or imported for the AP1 test case, vendors must submit the corresponding statistics. To do so, go to the AP1 Data Sheet, and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e., test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results after processing the file as described below.

Screenshots

For each Saved View in the model, provide one screenshot, which illustrates the layout (displayed geometry and annotations, model orientation, and zoom factor). Give the name of the view as the description of the screenshot. Note that CASEAR allows the addition of multiple screenshots per dataset.

Note that in order to count the GD&T elements for the statistics, per agreement during the R22J Review Meeting, the actual STEP entity types (datum, datum_target, etc.) shall be considered.

Note that all statistics – native and target – shall be based on the Semantic PMI Representation data only, and not take any presentation into account.

Data Sheet Columns

column name	description
model	The name of the test model, here 'ap1'
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'stp'
assem_struct	pass/fail - if the model structure (assembly tree) was transferred correctly, i.e. no nodes have been added or removed, and all elements are on the correct hierarchical level.
fref_found	all/partial/none - indicates if all, some or none of the references to the external files can be found in the assembly structure file(s), and if they are correctly associated with the respective nodes in the assembly structure.
eer_found	all/partial/none – indicates if all, some, or none of the External Element References could be resolved, i.e., if the intended target elements were found in the referenced files.
num_assem_pmi	The total number of assembly-level (root node) semantic PMI defined in the model.
num_assem_views	The number of Saved Views defined at assembly level (root node).
valid_assem_pmi	all/partial/none - indicates whether all, some or none of the assembly-level PMI are defined correctly (per test case definition and Recommended Practices).
valid_pmi_struct	all/partial/none – indicates whether all, some or none of the assembly-level PMI are linked correctly with the assembly structure, i.e., target the correct components via ComponentPathShapeElement.
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.2 Test Case CO2: Composite Materials (Ply Contour, EEOP & MEOP)

All information about this test case can also be viewed in CAESAR on its Information page.

2.2.1 Motivation

For several years, some STEP composite interfaces have been available in several CAD tools such as CATIA V5, FiberSIM and in CT CoreTechnologie tools, with a certain level of maturity proven by LOTAR pilot projects.

The goal of including Composite Materials in a CAx-IF test round is to align these implementations and provide an official framework for composite materials implementation tests as STEP AP 242 Ed.2 since it includes this capability.

2.2.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#41	Composite Validation property at part level
#42	Composite Validation property for each Laminate table, Sequence, Ply, core, ply piece, rosette

#44	Composite EEOP & MEOP
#46	Composite Core Samples
#77	Composite ply shape explicit contour
#78	Ply Material identifier

2.2.3 Approach

The scope of this test case is the “exact implicit” representation of composites where the ply geometry is based on surfaces and contours. “Basic” composite validation properties at the part level are also in scope of this test case. The approximate explicit representation of composite plies, where there is a 3D tessellated solid for each ply, is out of scope for this test case.

In addition, the tests in Round 57J consider the Engineering Edge of Part (EEOP) and Manufacturing Edge Of Part (MEOP) definitions given in the test model. The EEOP denotes the dimensions of the finished part, while the MEOP denotes the boundary to be used for manufacturing the laminate.

The approach is to export and to import composite information in STEP AP242 based on:

- AP242 Edition 3 or Edition 4 IS Longform Express Schema
  - Available on the MBx-IF homepage under “MBx > Resources > EXPRESS Schemas”.
- Recommended Practices for Composite Materials; Draft Version 4.3; 2 October 2025
  - Available on the MBx-IF homepage under “CAx > Rec. Practices”
- Draft Recommended Practices for Composite Structure Validation Properties; Release 0.30; 20 October 2025
  - Available on Nextcloud under “MBX-IF > CAX-IF > Draft Recommended Practices”.

Note: As the validation properties recommended practices have not been completely agreed upon, some tests will be done by end user checks.

2.2.4 Testing Instructions

The native model is the file "CPD_PUBLIC_LOTAR.CATPart" which is available as *co2_native.zip* on Nextcloud, folder:

MBX-IF > CAX-IF > CAX-IG > Round 52J > CO2_CO5

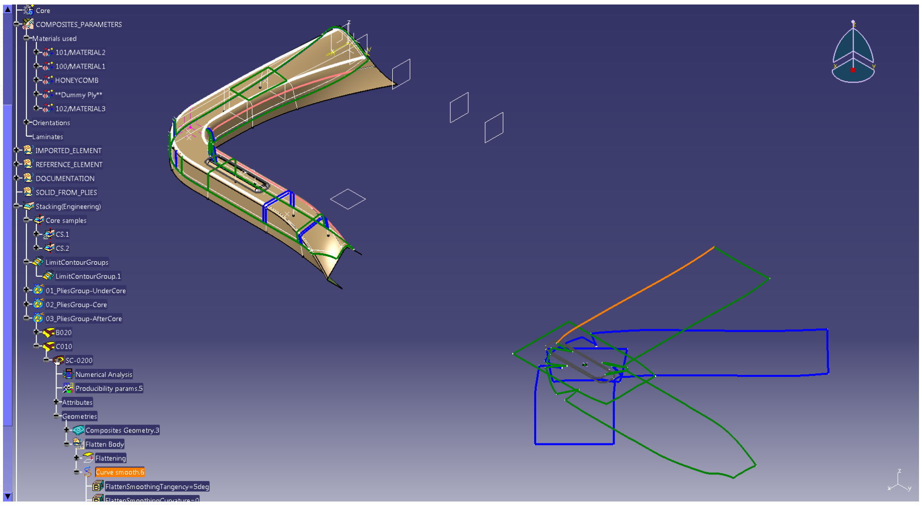


Figure 3: Illustration of the CO2 Test Case

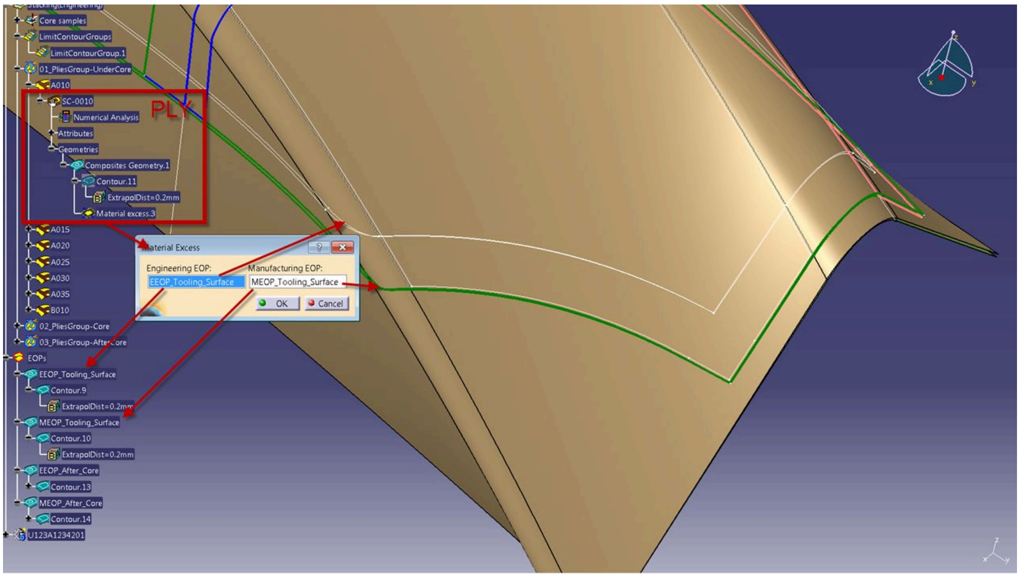


Figure 4: CO2 Details for EEOP & MEOP

2.2.5 Statistics

For each STEP file exported or imported for the CO2 test case, vendors must submit the corresponding statistics. To do so, go to the [CO2 Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e., test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a file, report the results found after processing the file as described below:

Ply-related Statistics

Several of the Statistics for this test case are related to a specific ply within a specific sequence (e.g., material, orientation, rosette). The statistics cannot evaluate this for all plies in the model. Hence, the idea is to select one specific (interesting) sequence and ply on export, and to publish its name in the "Composite Ply Sequence" field of the statistics. Then, fill in the other ply-related statistics with the values as valid for this particular sequence and ply. After import, select the sequence and ply with the name given in the native statistics, and again provide the values valid for this particular sequence and ply.

The sequence and ply to be used for evaluating the CO2 test case in Round 57J is:

`PLY SC-0035 of SEQUENCE A035`

Statistics for Core Sample Point

The position of the point for the Core Sample shall be given for:

`CORE SAMPLE CS1`

Statistics for Flatten Pattern

The length of the curve contour of the flatten pattern shall be given for:

`PLY SC0200 of SEQUENCE C010`

Data Sheet Columns

These statistics will be enhanced in future test rounds, especially with the release of newer versions of the Recommended Practices for Composite Structure Validation Properties.

column name	description
model	The name of the test model, here 'CO2'
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'stp'
unit	The unit the model is designed in
compos_tables	The number of Composite Tables in the Model
sequences	The number of Sequences in the model
plies	The total number of plies in the file
num_materials	Total number of Materials defined
compos_table_name	The name of the Composite Table of the model
ply_sequence	The ID of the Sequence and the ID of the Ply within that Sequence for all ply-related statistics; e.g., "Ply.P4 of Sequence.S4".
seq_ply_number	The total number of Plies defined within the Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
seq_ply_material	The name of the Material of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
seq_ply_mat_type	The type of Material of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
seq_ply_orient	pass/fail - whether the orientation of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet was correct
seq_ply_rosette	The name of the Rosette of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
ply_surface_area	The value of the area of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
cores	The total number of cores in the file
fp_length	The length of the curve contour of the Flatten pattern of the ply and sequence indicated in the Test Suite document.
validation_c_tables	Total number of Composite Tables in the model, as received via the validation properties capability
validation_se-quences	Total number of Sequences as received via the validation proper-ties capability
validation_plies	Total number of Plies (entire assembly) as received via the valida-tion properties capability
validation_c_materi-als	Total number of Materials as received via the validation properties capability
validation_c_orient	pass/fail, indicates whether the Number of Orientations per part in the model matches the Composite validation property value given in the STEP file

column name	description
validation_ply_area	pass/fail, indicates whether the sum of all ply surface areas in the part matches the Composite validation property value given in the STEP file
validation_ply_centroid	pass/fail, indicates whether the sum of all ply geometric centroids in the part matches the Composite Validation Property value given in the STEP file
valid_cvp	pass/fail, is the instantiation of the validation properties for Tessellated Geomtry in the STEP file as per the recommended practices?
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.3 Test Case CO3: Composite Materials (3D Explicit Ply Representation)

All information about this test case can also be viewed in CAESAR on its Information page.

2.3.1 Motivation

For several years, some STEP composite interfaces have been available in several CAD tools such as CATIA V5, FiberSIM and in CT CoreTechnologie tools, with a certain level of maturity proven by LOTAR pilot projects.

The goal of including Composite Materials in a CAx-IF test round is to align these implementations and provide an official framework for composite materials implementation tests as STEP AP242 includes this capability.

2.3.2 Approach

The scope of this test case is the “3D tessellated” representation for each ply. The approximate explicit representation of composite plies includes a 3D tessellated solid for each ply.

The approach is to export and to import composite information in STEP AP242 based on:

- AP242 Edition 3 or Edition 4 IS Longform Express Schema
    - Available on the MBx-IF homepage under “MBx > Resources > EXPRESS Schemas”.
- Recommended Practices for Composite Materials; Draft Version 4.3; 2 October 2025
    - Available on the MBx-IF homepage under “CAx > Rec. Practices”
- Draft Recommended Practices for Composite Structure Validation Properties; Release 0.30; 20 October 2025
    - Available on Nextcloud under “MBX-IF > CAX-IF > Draft Recommended Practices”.

2.3.3 Testing Instructions

The test case ASME_Y14.37_RosetteType2.CATPart will be used. The model has been provided by The Boeing Company.

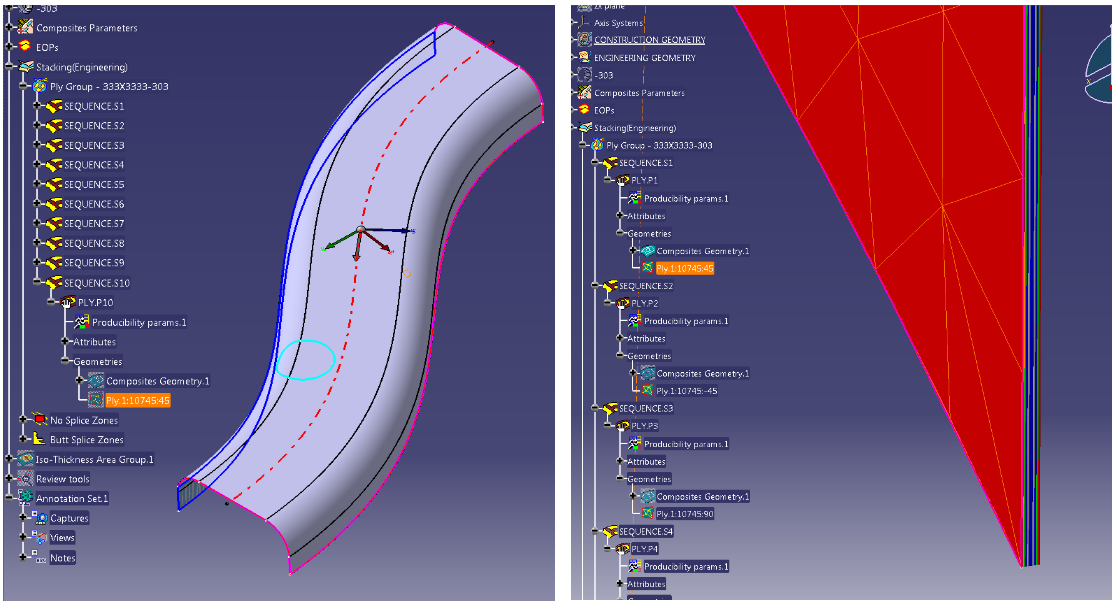


Figure 5: Illustration of the CO3 Test Case

The test model contains the 3D tessellated representation of each ply.

The test case is available on Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 57J > CO3

2.3.4 Statistics

For each STEP file exported or imported for the CO3 test case, vendors must submit the corresponding statistics. To do so, go to the [CO3 Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e. test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results found after processing the file as described in the table below.

Ply-related Statistics

Several of the Statistics for this test case are related to a specific ply within a specific sequence (e.g., material, orientation, rosette). The statistics cannot evaluate this for all plies in the model. Hence, the idea is to select one specific (interesting) sequence and ply on export, and to publish its name in the "Composite Ply Sequence" field of the statistics. Then, fill in the other ply-related statistics with the values as valid for this particular sequence and ply. After import, select the sequence and ply with the name given in the native statistics, and again provide the values valid for this particular sequence and ply.

The sequence and ply to be used for evaluating the CO3 test case in Round 57J is:

`PLY.P4 of SEQUENCE S.4`

Data Sheet Columns

column name	description
model	The name of the test model, here 'CO3'
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'stp'
unit	The unit the model is designed in
compos_tables	The number of Composite Tables in the Model
compos_table_name	The name of the Composite Table of the model
sequences	The number of Sequences in the model
plies	The total number of plies in the file
num_materials	Total number of Materials defined
num_saved_views	The number of Saved Views defined in the model
num_annotations	The total number of Annotations defined in the model.
ply_sequence	The ID of the Sequence and the ID of the Ply within that Sequence for all ply-related statistics. For CO3, use: "Ply.P4 of Sequence.S4".
seq_ply_number	The total number of Plies defined within the Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
seq_ply_material	The name of the Material of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
seq_ply_mat_type	The type of Material of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
seq_ply_orient	pass/fail - whether the orientation of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet was correct
seq_ply_rosette	The name of the Rosette of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
ply_rosette_type	The type of the Rosette of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
ply_contour_area	The surface area of the ply contour of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
facets	The number of facets in the Tessellated model
ply_explicit_area	The surface area of the 3D explicit representation (tessellated geometry) of the specific Ply and Sequence as listed in the "Composite Ply Sequence" column of the data sheet.
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.4 *Test Case HTC: Hole Features*

All information about this test case can also be viewed in CAESAR on its Information page.

2.4.1 Motivation

Product Manufacturing Information (PMI) is required for numerous business use cases in the context of STEP data exchange. Among others, it is a prerequisite for long-term data archiving. In addition, PMI can be used to drive downstream applications such as coordinate measuring and manufacturing.

Within the area of PMI, the semantic definition of holes is a specific topic which has recently received heightened attention by the user community. In addition to the mere diameter of a cylindrical hole, this also includes information about specific bottom conditions, or links various data objects to create more complex features such as a counterbore.

Depending on the overarching use case, holes can be represented in various ways. The STEP data model provides a simplified as well as an explicit representation. Both are in scope of this test case.

To test a wide variety of holes, a dedicated Hole Test Case (HTC) has been created by NIST. Based on discussions at the Round 56J Review Meeting, a variant of the HTC model originally created for testing Persistent IDs (PDC) will be used for testing hole features in R57J.

2.4.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#4	Simplified Hole (Native function and STEP conversion)
#97	Explicit Hole

2.4.3 Approach

The approach to be used for the hole features in the HTC model is described in the draft "Recommended Practices for Hole Information" (Version 0.1, 31 March 2021), which can be found on Nextcloud, folder:

MBX-IF > CAX-IF > Draft Recommended Practices

The general PMI information shall adhere to the "Recommended Practices for the Representation and Presentation of PMI (AP242)", (Version 4.1, 20 June 2024), which is available on the public MBx-IF homepage under "CAx > Rec. Practices".

In addition to the usual PMI scope, the following functionalities are specific to the HTC:

- Correct implementation of round hole
- Correct implementation of counterbore hole
- Correct implementation of countersink hole
- Correct implementation of counter drill hole

The AP242 schema to be used is the AP242 Edition 3 schema, which is available on the MBx-IF homepage under "Resources → EXPRESS Schemas". This schema provides full support of the latest changes and additions in the Recommended Practices.

Pre-checking of files with SFA: All vendors generating STEP files for the HTC test case shall run them through the latest version of NIST's STEP File Analyzer and Viewer (SFA; currently version 5.30). The tool provides feedback on basic syntax errors such as missing or malformed entity instances. Files with such errors will not be accepted for testing.

SFA can be downloaded for free from the NIST homepage.

2.4.4 Testing Instructions

2.4.4.1 Test Model Overview

The test model to be used for the HTC test case is the PDC model, which is available on Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 57J > HTC

Note 1: The CATIA native file has been updated compared to previous rounds, to align the depth of the countersink holes.

Note 2: The PDC Test Case uses the STC-9 model.

The test case contains holes modeled in a variety of ways on various surface conditions as well as complex holes including counterbore, countersink, and counter drill. A 3D PDF document is provided showing the individual Saved View configurations.

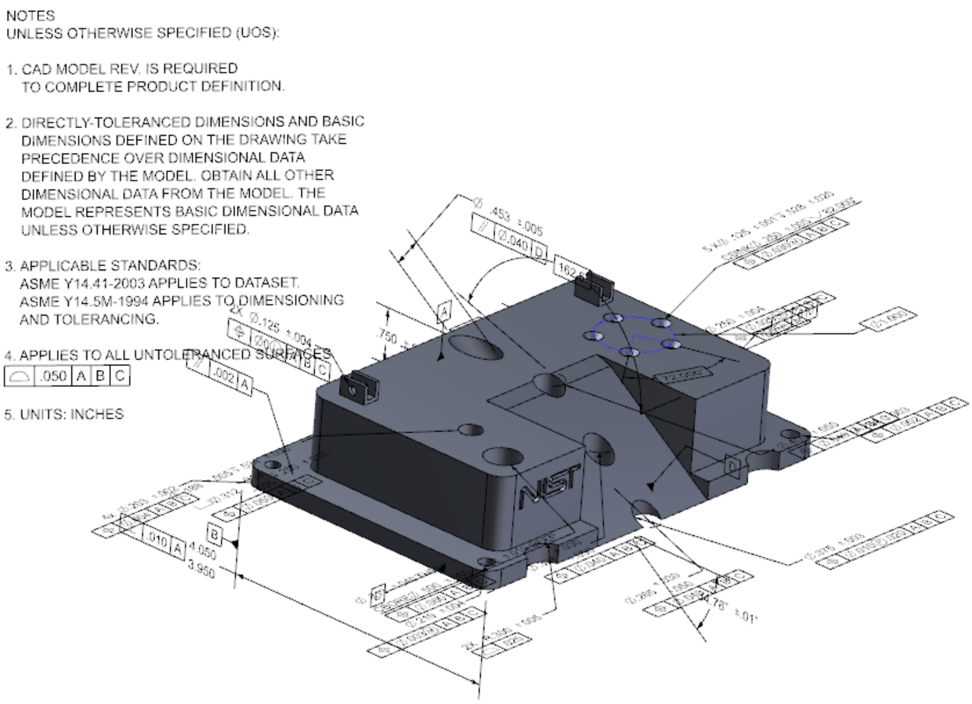


Figure 6: View of the PDC model for the HTC test case with all annotations switched on

2.4.4.2 Test Model Configuration

As the HTC test case is, at its core, a PMI test case, all the test model configuration considerations from the JPMI test case (see section 2.5.5) apply here as well. In short, the model shall contain:

- Semantic PMI Representation
- Tessellated PMI Presentation
- [Optional] PMI Presentation Placeholders
- Saved Views
- Correct linking between graphics, semantics, and part geometry
- [Optional] Validation Properties
- [Optional] Persistent IDs, including on semantic holes

In addition, specifically to this test case, the model shall contain:

- Semantic Hole Definitions – Holes shall be defined as features, following the Recommended Practices for Hole Information. More complex features shall be represented using the specific types, e.g. a counterbore hole definition, which links the various constituent features, as far as applicable. This higher level of semantics will allow downstream applications to identify such features more easily than just having stacked cylinders with dimensions.

2.4.5 File Naming Convention and SFA Checking

In order for SFA to correctly identify the test cases, the STEP files must strictly follow the following naming convention:

- `nist-htc-systemcode-242.stp`

For instance, `nist-htc-nx-242.stp` would be the STEP file exported by Siemens out of NX for the HTC model.

2.4.6 Statistics

For each STEP file exported or imported for an HTC test case, vendors must submit the corresponding statistics. To do so, go to the HTC Data Sheet, and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e., test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results after processing the file as described below.

Screenshots

For each Saved View in the model, provide one screenshot, which illustrates the layout (displayed geometry and annotations, model orientation, and zoom factor). Give the name of the view as the description of the screenshot. Note that CASEAR allows the addition of multiple screenshots per dataset.

Note that all statistics – native and target – shall be based on the Semantic PMI Representation data only, and not take any presentation into account.

Note that for evaluation, the spreadsheets generated by the STEP File Analyzer and Viewer will be amended with corresponding aggregations of relevant counts and charts.

HTC Data Sheet Columns

column name	description
model	The name of the test model, here: 'htc'
system_n	The system code of the CAD system creating the STEP file.
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'stp'
angular_dimensions	The number of basic dimensions processed
basic_dims	The number of basic dimensions processed
bilateral_tols	The number of Dimensions that have a Bilateral Tolerance.

column name	description
limit_tols	The number of dimensions that show the upper and lower limits.
round_holes	Total number of explicit or basic round holes (Hole Rec. Pracs. 5.1.2).
counterbore_holes	Total number of counterbore holes (Hole Rec. Pracs. 5.1.3).
countersink_holes	Total number of countersink holes (Hole Rec. Pracs. 5.1.3).
counterdrill_holes	Total number of counterdrill holes (Hole Rec. Pracs. 5.1.4).
num_pid_sem_hole	The number of semantic text PMI elements processed with persistent IDs
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.5 Test Case JPMI: Graphic & Semantic PMI using JAMA/JAPIA Models

All information about this test case can also be viewed in CAESAR on its Information page.

2.5.1 Motivation

Product Manufacturing Information (PMI) is required for numerous business use cases in the context of STEP data exchange. Among others, it is a prerequisite for long-term data archiving. In addition, PMI can be used to drive downstream applications such as coordinate measuring and manufacturing.

Semantic PMI Representation relates to the capability to store PMI data in the STEP file in a computer-interpretable way, so that it can be used for model redesign or downstream applications. Though the definition of the data is complete, it is by itself not visible in the 3D model.

In addition to use cases that require a fully defined, precise, semantic definition of the part geometry and associated PMI, there are also scenarios where the presentation of the data - geometric elements and annotations - for visual consumption are the primary goal. In such cases, a simplified and optimized version of the model is sufficient. The tessellated geometry model included in AP242 provides an efficient mechanism to support this.

In addition to the well-known suite of test models from NIST, which have been tested for many years, the Japanese industry has conducted a comprehensive testing activity similar to the original NIST MBE PMI Validation and Conformance Testing Project. In total, JAMA and JAPIA have created seven test models - four single-part models and three assembly-models - supporting different industry use cases, each available in four different native formats. These models have also undergone extensive review and testing. Due to their increased complexity, they are the logical next models to be used for PMI testing.

2.5.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#49	Saved views Validation Properties
#124	Default saved view
#184	Annotation placeholder

2.5.3 Approach

The approach to be used is described in the "Recommended Practices for Representation and Presentation of PMI (AP242)", (Version 4.1, 20 June 2024) which can be found on the public MBx-IF homepage under "CAx > Rec. Practices".

Within the PMI area, the following functionalities are in scope of Round 57J:

- Semantic PMI Representation
- Tessellated PMI Presentation
- Correct implementation and definition of the Saved Views (view layout and contents)
- Linking of PMI Representation to Presentation
- Transfer of editable PMI text as User Defined Attributes
- PMI Validation Properties (Representation & Presentation)
- Presentation Placeholder (including Placeholder Leader Lines if supported)

The AP242 schema to be used is the final AP242 Edition 4 schema, which is available on the public MBx-IF homepage, under

MBx > Resources > EXPRESS Schemas

This schema provides full support of the latest changes and additions in the Recommended Practices, in particular, the Presentation Placeholder.

Pre-checking of files with SFA: All vendors generating STEP files for the PMI test case shall run them through the latest version of NIST's STEP File Analyzer and Viewer (SFA; currently version 5.35). The tool provides feedback on basic syntax errors such as missing or malformed entity instances. Files with such errors will not be accepted for testing.

For more information on SFA and download links, see

https://www.mbx-if.org/home/cax/resources/sfa/

2.5.4 Testing Instructions

2.5.4.1 JAMA / JAPIA Test Model Overview

The JAMA / JAPIA test models can be found on Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 54J > JAMA_JAPIA

The folder contains one ZIP file for each model, containing all the different native models. Each model is available in the following formats:

- CATIA V5 R31
- 3Dx 2022X
- NX 2015
- Creo 7.0

These are the same models as tested in Round 56J. JAMA is aware of requests to update these models based on testing feedback but have prioritized the completion of a new complex model before revisiting the existing ones.

A PDF document with detailed information about each model, as well as the activity behind creating them, is also provided in the same folder

- See file *240620_JAMAJAPIA_SampleDataExplanation_ en_09a.pdf*

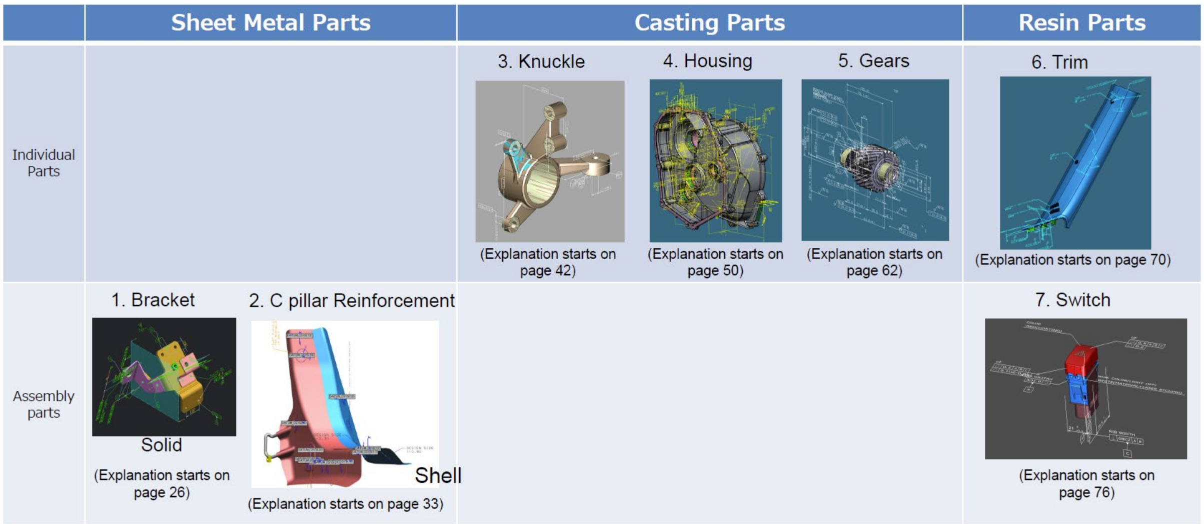


Figure 7: Overview of the set of JAMA / JAPIA test models

2.5.4.2 JAMA / JAPIA Test Model Selection

For testing in Round 57J, the four single-part models will be used:

- Knuckle *
- Housing
- Gears
- Trim *

<u>*** Note:**</u> The packages for these files contain multiple parts and an assembly definition. See the provided “Sample Data Explanation” presentation for details.

On the “Knuckle” model in particular, it was agreed that there are three possible choices:

- Use all of the three parts (convey all the intents of JAMA/JAPIA)
- Use either of the solid shape + "40000_11111_3DA" (convey design intents of the part, but in a strange formation...)
- Use only "40000_11111_3DA" (Pure PMI test, but design intents are totally lost)

I.e., testing the “3DA” part is mandatory, including the structure and any of the other shapes is optional, though recommended. This unusual model setup was specifically requested by Japanese industry representatives.

2.5.5 Test Model Configuration

The following functionality shall be included in the test files provided for this round of testing, as far as it has been implemented by the CAx-IF participants and is described in the Recommended Practices:

- <u>PMI Representation</u> – the re-usable representation of PMI data should be included in all PMI models to the extent supported by the native system.
- <u>PMI Tessellated Presentation</u> – Many CAD systems require some minimal presentation information to be able to handle the PMI data in a model. Usually, both PMI representation and presentation data are included in the same file. Thus, some form of presentation information shall be included in the PMI test case as well.
- <u>PMI Presentation Placeholder</u> – This information enables a target system with PMI authoring capability to recreate the presentation of a PMI element based on its Semantic Representation data. It intends to provide a minimal set of presentation information to

CAD systems, which require information such as the leader line attachment point on the part geometry to create the corresponding Semantic PMI Representation elements.

  - Implementation of this capability requires AP242 Ed.3, as well as the approach from section 7.2 of the PMI Recommended Practices (Version 4.1).

- Definition of "Saved Views" – as far as supported, include the saved views defined in the models, which contain a subset of annotations in the file, and provide a pre-defined position of the model in the design space.
  - All models have multiple Saved Views defined. In the test case definition documents, each page of the PDF document represents one Saved View.
  - For each view, a screenshot showing the model layout (displayed elements, orientation, zoom) shall be provided.
  - **Note** that it is possible to attach several screenshots to one set of statistics in CAESAR. The name of the view shall be given as description for the screenshot.
  - Saved Views shall correctly show (or hide) the part geometry, as well as the non-solid Supplemental Geometry contained in some of the models (see section 9.4.2 / Figure 96 in the PMI Rec. Practices v4.1).
- Editable PMI Text – Some information relevant for PMI is not encoded in semantic entities, but given as plain text, such as the title block information or additional text on feature control frames. In the context of semantic data exchange, this content needs to be editable in the target system. The approach to be used for this is based on the transfer of User Defined Attributes, and its application in the context of PMI is described in section 7.4 of the PMI Recommended Practices v4.1.
- Linking PMI Representation to Presentation – If a model contains PMI Representation information as well as Presentation data, the corresponding elements shall be linked together, so that a Representation element "knows" which annotation it is being presented in the model. The approach to create this link is described in section 7.3 of the PMI Rec. Pracs. (v4.1).
- Cross-highlighting of annotations and annotated shape – if supported, include in the STEP file the information necessary to maintain the association between graphic annotations and the annotated shape elements in a way, that after import, when highlighting an annotation, the shape elements annotated by it are highlighted too, and vice versa.
- Validation Properties – All participants providing STEP files for this test case are encouraged to include validation properties for PMI semantic representation and graphic presentation, as defined in the PMI Recommended Practices v4.1, sections 10.1 and 10.2 respectively.

Also refer to the "Sample Data Explanation" presentation for test model translation configuration considerations.

2.5.6 File Naming Convention and SFA Checking

In order for SFA to correctly identify the test cases, the STEP files must strictly follow the following naming convention:

- `jpmi-modelname-systemcode-242.stp`

For instance, `jpmi-knuckle-do-242.stp` would be the STEP file exported by Datakit out of Creo for the "Knuckle" model.

2.5.7 Statistics

For each STEP file exported or imported for the JPMI test case, vendors must submit the corresponding statistics. To do so, go to the JPMI Data Sheet, and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e., test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results after processing the file as described below.

Screenshots

For each Saved View in the model, provide one screenshot, which illustrates the layout (displayed geometry and annotations, model orientation, and zoom factor). Give the name of the view as the description of the screenshot. Note that CASEAR allows the addition of multiple screenshots per dataset.

Note that in order to count the GD&T elements for the statistics, per agreement during the R22J Review Meeting, the actual STEP entity types (datum, datum_target…) shall be considered.

Note that all statistics – native and target – shall be based on the Semantic PMI Representation data only, and not take any presentation into account.

Note that for evaluation, the spreadsheets generated by the STEP File Analyzer and Viewer will be amended with corresponding aggregations of relevant counts and charts.

Data Sheet Columns

column name	description
model	The name of the test model, here 'jpmi', with one of the following suffixes: 'knuck', 'trim', 'gears', 'hous'
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'stp'
scope	A short designation for the contents of the model as defined in the Test Suite. This is for information only; there will be no results for this field.
dimensions	The number of dimensions processed
datums	The number of datums processed
datum_targets	The number of datum targets processed
tolerances	The number of tolerances (all types combined) processed, regardless of composition.
compos_tols	The number of composite tolerances processed (number of instances of geometric_tolerance_relationship per section 6.9.9. in the PMI Rec. Pracs. v4.1).
labels	The number of labels processed
pmi_semantic_txt	all/partial/none – whether 'semantic' (editable) PMI text was transferred correctly (content and associativity)

column name	description
pmi_semantic_val-prop	all/partial/none – whether the validation properties for Semantic PMI Representation matched for all, some or none of the semantic PMI elements.
saved_view	The name of the Saved View which is the basis for the view-related statistics
view_annot	The number of annotations (polyline or tessellated) included in the specified saved view. This does NOT include annotation placeholders.
view_placeholders	The number of annotation placeholders included in the specified saved view.
view_pos	pass/fail, whether the model orientation and zoom factor stored for the Saved View could be restored successfully.
elem_visibility	all/partial/none – whether all, some, or none of the elements to be displayed in the indicated saved view were mapped correctly into the corresponding draughting_model.
pmi_savedview_valprop	all/partial/none - whether the validation properties for PMI Saved Views matched for all, some or none of the views defined in the model.
pmi_graphic_pres	all/partial/none – whether the graphic PMI annotations included in the file could be processed correctly
pmi_present_valprop	all/partial/none – whether the validation properties for Graphic PMI Presentation matched for all, some or none of the presentation elements.
pmi_linked_pres_rep	all/partial/none – whether the Semantic PMI Representation elements and (Graphic) PMI Presentation elements were linked correctly together.
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.6 *Test Case KM4: Kinematics*

All information about this test case can also be viewed in CAESAR on its Information page.

2.6.1 Motivation

CAD methods have been used for many years now to design individual parts and assemblies of all sizes across all industries, from a single rivet to an entire airplane. Classically, the main focus is to ensure that the part can be manufactured correctly.

Products such as cars or planes are not static, however, contain many moving components as well: engine, power windows, foldable roof, windshield wipers, cargo doors, etc. Thus, Kinematics are used to ensure they move correctly, and also to illustrate the behavior of the finished product. The use cases range from the definition of the Kinematic Mechanism, providing all relationships and constraints between the elements so that their definition can be changed in the receiving application, to Kinematic Motion, which works like a movie by providing discrete positions of the components over time.

The goal is to use a neutral standard format – AP242 Domain Model XML – for the definition of the Kinematic mechanisms and motion, with external references to the applicable geometry format for the respective use case.

After the high-order kinematic pairs, which require external element references (EER) to specific geometric elements inside the referenced geometry file, were added in Round 55J. Round 56J extended the scope yet again by including the so-called "Structure" models, which include test cases with mechanisms on several assembly nodes, use of several occurrences of the same part within a mechanism, and closed kinematic chains. With this latest addition, the scope of available unit test cases for Kinematics is fully covered, which will allow vendors to extend the coverage of their translators step by step.

2.6.2 Approach

The approaches for Kinematic Mechanism as well as for External Element References are described in the following documents, which can be found on the public MBx-IF homepage under **CAx > Rec. Practices**:

- Rec. Practices for AP242 Domain Model XML Product & Assembly Structure
    - v4.0 (20 Oct 2025)
- Rec. Practices for AP242 Domain Model XML Kinematics
    - v1.3 (20 Oct 2025)
- Presentation “EER use cases PDM, PMI and Kinematics”
    - v0.19 (see **MBX-IF > CAX-IF > Draft Recommended Practices** on Nextcloud).

Implementation of the External Element References:

- Domain Model XML: For the outgoing references, the approach supported by AP242 Ed.4 (Part 15 Ed.2) shall be used.
- MIM Part 21: For the incoming references, UUIDs in the Data Section shall be used (i.e., no Anchors)

The tests in Round 57J shall use the final AP242 Edition 4 XSD and EXP schemas, which can be found on the public MBx-IF homepage under **MBx > Resources**.

Collaboration with JT-IF

The Kinematics capabilities are being developed and tested in close collaboration with the JT Implementor Forum. While the file format for the part geometry is different, the AP242 XML files and the Kinematics definitions therein are identical. Test files for this capability will be exchanged between the two groups, in order to increase the number of participating systems. Testing feedback will be exchanged as well between the actively participating vendors, and any resulting improvements will be documented in the joint Recommended Practices. These collaborative efforts are coordinated by Jochen Boy (jochen.boy@prostep.com).

2.6.3 Testing Instructions

Round 57J continues using the suite of KM4 models. This is a set of unit test cases, where each model focuses on a single joint type or structure, allowing vendors to focus on the development of specific capabilities without the side-effects introduced by more complex industry models. In their entirety, the set of KM4 models covers the full scope of the Kinematics Recommended Practices.

There are 15 models which represent the categories “Low Order” and “Low Order with Motion Coupling”. In addition, eight more models for “High Order” kinematic pairs were added in Round 55J. These will require the implementation of External Element References (EER) from the AP242 XML file with the assembly and kinematics definitions to the respective geometry files and the target elements therein.

The final scope extension was the addition of the "Structure" models in Round 56J..

There are 28 models in total:

- 10x **LowOrder** (Cylindrical, Fully Constrained, Homokinetic, Homokinetic Revolute, Planar, Prismatic, Revolute, Spherical, Universal, Universal Revolute)

- 5x **LowOrder with Motion Coupling** (Cable, Gear, Gear 90°, Rack and Pinion, Screw)
- 8x **HighOrder** (Point on Curve (Circle, Circle Point, Spiral, Spiral Point), Point on Surface (Surface, Surface Point), Roll on Curve, Slide on Curve)
- 5x **Structure** (Chain, Groups, MultipleOccurences, SpecifiedOccurences, Structure)

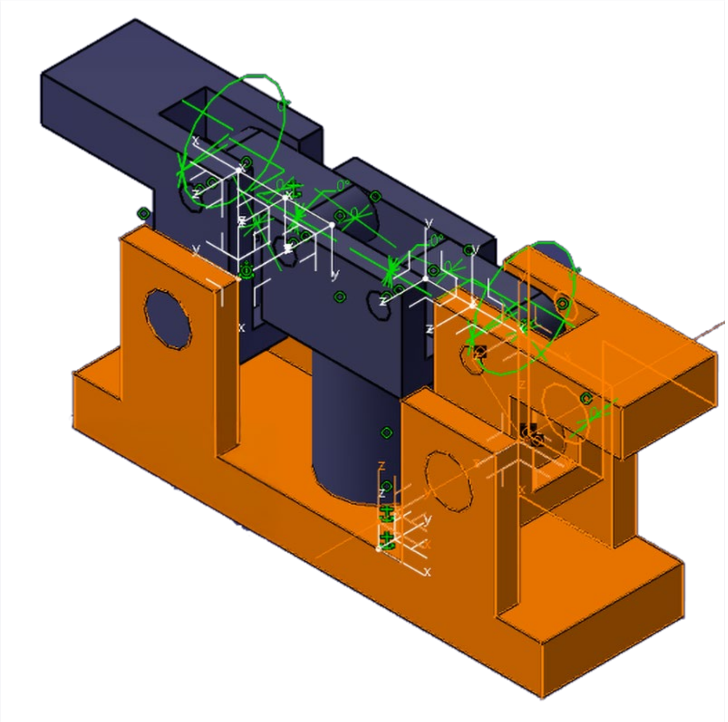


Figure 8: Illustration of the KM4 model for "SpecifiedOccurences"

One model has been created for each type of kinematic pair. Most models also contain additional kinematic pairs to build a working mechanism. This includes different variants for certain joint types.

The native CAD files (CATIA V5-6 R2024 and NX 2306) for all KM4 test models can be found on Nextcloud:

MBX-IF > CAX-IF > CAX-IG > Round 56J > KM4

In addition, this folder also contains a PowerPoint presentation illustrating the different models. There are a few known limitations, e.g., different representations in the authoring system, etc. These are documented in the slides.

Regarding the testing scope, the following has been agreed:

- **Kinematic Mechanism** is the primary use case, and the corresponding definitions shall be included in all provided files.
- **Kinematic Motion** can be added by anyone interested in supporting this extended scope.
- **Assembly & Kinematic Data** shall be provided in a single AP242 Domain Model XML file, using the schema indicated above.
- **Geometry** shall be included as AP242 Part 21 files.

2.6.4 Statistics

For each STEP file exported or imported for the KM4 test case, vendors must submit the corresponding statistics. To do so, go to the [KM4 Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select 'full support' (i.e., test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a file, report the results found after processing the file as described below.

Kinematics-specific Statistics

For more detailed information about the Kinematics-specific statistics, please refer to section 4.12 of the Kinematics Recommended Practices mentioned above.

In order to distinguish between the different models with the suite of models tested under the KM4 designator, the following suffixes shall be used in the 'model' field of the statistics:

suffix	Kinematic Pair	Kinematic Pair Type
KM4_cyl	Cylindrical	Low Order
KM4_cons	Fully Constrained	Low Order
KM4_hkin	Homokinetic	Low Order
KM4_hkrev	Homokinetic Revolute	Low Order
KM4_plan	Planar	Low Order
KM4_prism	Prismatic	Low Order
KM4_rev	Revolute	Low Order
KM4_spher	Spherical	Low Order
KM4_univ	Universal	Low Order
KM4_unrev	Universal Revolute	Low Order
KM4_cable	Cable	Low Order with Coupling
KM4_gear	Gear 0°	Low Order with Coupling
KM4_gr90	Gear 90°	Low Order with Coupling
KM4_rap	Rack and Pinion	Low Order with Coupling
KM4_screw	Screw	Low Order with Coupling
KM4_pocc	Point on Curve Circle	High Order
KM4_poccp	Point on Curve Circle Point	High Order
KM4_pocs	Point on Curve Spiral	High Order
KM4_pocsp	Point on Curve Spiral Point	High Order
KM4_pos	Point on Surface	High Order
KM4_posp	Point on Surface Point	High Order
KM4_roc	Roll on Curve	High Order

suffix	Kinematic Pair	Kinematic Pair Type
KM4_soc	Slide on Curve	High Order
KM4_chain	Chain	Structure
KM4_grps	Groups	Structure
KM4_multi	MultipleOccurences	Structure
KM4_spec	SpecifiedOccurrences	Structure
KM4_struc	Structure	Structure

Data Sheet Columns

column name	description
model	The name of the test model, here 'KM4'
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'jt'
assem_struct	pass/fail - if the model structure (assembly tree) was transferred correctly, i.e. no nodes have been added or removed, and all elements are on the correct hierarchical level.
num_mech_pairs	The number of low/high order Kinematic Pairs defined for a Kinematic Mechanism.
kin_placements	The number of AxisPlacements used by KinematicPairs
valid_kin_pair	pass/fail - Whether the KinematicPair in focus for the specific test case was exported and retrieved correctly.
kin_mech_acts	The number of Kinematic Pairs that have a non-zero value in the attribute `actuation`, i.e. where an initial movement can occur.
kin_limits	pass/fail, if the lower and upper limits defined for kinematic pairs were transferred correctly.
kin_mech_valprops	all/partial/none - whether the validation properties for Kinematic Mechanism matched for all, some or none of the Kinematic definitions.
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.7 Test Case PDC: Persistent IDs for Downstream Consumption

All information about this test case can also be viewed in CAESAR on its Information page.

2.7.1 Motivation

The ability to track a product's model information during design iteration, and from design iteration through to manufacturing and quality analysis has been limited by the lack of support for persistent IDs in STEP.

With the inclusion of persistent IDs collaborating systems should now be able to exchange model data, enabling evaluation of that data downstream. This suggests the ability to retain IDs contained in external data from a sender and reference those entities by the receiver. When a change to that model data occurs on the sender's side, the receiver should be able to update the receiver's copy of that external data and have any dependent data in their own models that refer to that external change, and update to respond to the change.

As in the case of design iteration (see Test Case PDI), the ability to track model entities via Persistent IDs, will also allow downstream systems to update their representations of the design model and update their manufacturing and metrology planning to reflect changes in the design.

An additional benefit of the establishment of persistent IDs in STEP is the ability to retain a permanent audit trail of custody and connection between design and downstream systems for potential forensic analysis of critical product systems after in-service failure.

As stated earlier, vendors may choose to support either preprocessing native models to generate STEP data or postprocessing such STEP models. Post-processing will be performed to exercise the downstream consumption use case.

Finally, although not covered in this test case, the introduction of persistent IDs provides the ability of any contributor to the information stream associated with a product's lifecycle to add information to the model that can be connected to existing model content and that additional information can be retrieved by subsequent users and used as feedback from the contributor.

2.7.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#391	PID Designer requirement
#392	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (initial use).
#413	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (first full process flow)
#415	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (revision capture/impact 1).
#417	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (revision capture/impact 2).

2.7.3 Approach

The approach to be used is described in the "Recommended Practices for Permanent Entity IDs for Design Iteration and Downstream Exchange" (Version 1.7; 29 October 2025), which can be found on the public MBx-IF homepage under "CAx > Rec. Practices."

The approach to be used for downstream consumption is described in the "Recommended Practices for Cross-Domain Exchange" (Draft Version 1.0; 17 December 2025), which can be found on the MBx-IF Nextcloud, folder

> **MBX-IF > CAX-IF > Draft Recommended Practices > Rec_Prac_for_Cross_Domain_Exchange**

Within the domain of Persistent IDs, the following functionalities are in scope of Round 57J:

- Persistent IDs on Model (Product and Model Version) for
    - testing the retention of model ID information during exchange with downstream systems
- Persistent IDs on Geometry and Topology for
    - testing the retention of persistent IDs on geometry and topology with downstream applications that reference that geometry and topology

  - this concept includes the introduction of Persistent IDs on shape_aspect, when needed, to collect individual geometry elements into logical groups such as complex hole geometries and/or the collection of groups of shape aspects of patterned geometric groups
  - When sending and receiving systems have differing topological structures, e.g. periodic or aperiodic cylindrical holes and edges, implementers will use the uuid_set_item for collecting these pairs and assign a single UUID to the pair.
- Persistent IDs on Semantic PMI Representation for
  - testing the assignment and exchange of persistent IDs on semantic PMI that references that geometry and topology for downstream consumption.
  - testing the assignment and exchange of persistent IDs on semantic PMI for dependent manufacturing planning that reference that semantic PMI.
  - testing the assignment and exchange of persistent IDs on semantic PMI on dependent metrology planning that reference that semantic PMI.
- Persistent IDs on Semantic Text (PMI) Representation for
  - testing the assignment and exchange of persistent IDs on semantic text used to represent global notes that likely will include “unless otherwise specified” requirements and their impact on dependent manufacturing planning that reference that semantic text.
  - testing the assignment and exchange of persistent IDs on semantic text used to represent global notes that likely will include “unless otherwise specified” requirements and their impact on dependent metrology planning that reference that semantic text.
- Persistent IDs on User Defined Attributes for
  - testing the assignment and exchange of persistent IDs on attributes and their values that may be used as reference data for dependent manufacturing planning.

The following are out of scope for Round 57J and may be considered in the future:

- Persistent IDs on Geometry and Persistent IDs on Semantic PMI Representation for testing assembly constraints referencing those geometries.

To support the concepts in scope of this test case, the final AP242 Edition 4 schema shall be used, which can be found via the public MBx-IF homepage under “MBx > Resources > EXPRESS Schemas”.

2.7.4 Testing Instructions

Based on discussions at the Round 56J Review Meeting, the NIST STC-09 test model will be used for the PDC test case in Round 57J. The rationale for this was that NIST STC-09 has been tested by the majority of CAD vendors and the PMI published from that model is considered stable. Thie eliminates variability in the PMI and thus allows to focus on the associated Persistent IDs.

2.7.4.1 Test Model Access

The latest version of NIST STC-09 can be obtained from the MBx-IF web site under “CAx > CAx Resources > NIST PMI Models.” The models are available in CATIA V5-6R2023, NX2027, Inventor 2024 and Creo 10 formats.

The test case is a single exchange with no iterations, focusing on the downstream consumption of the provided information.

2.7.4.2 Test Model Configuration

For details on the test model configuration, refer to the 3D PDF of the NIST STC-09 test model which is included when downloading the models via the link included above.

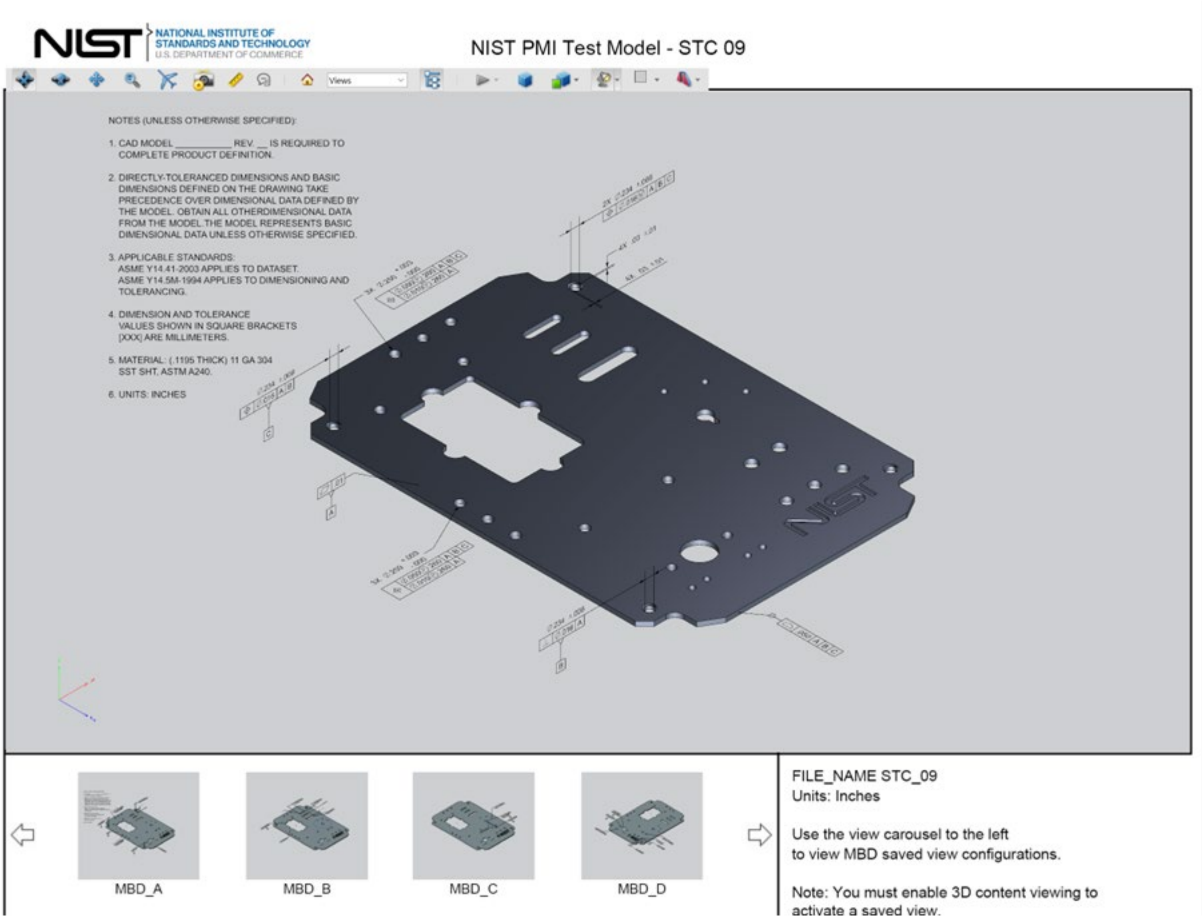


Figure 9: Illustration of the STC-09 model used for the PDC test case

2.7.4.3 File Naming Convention and SFA Checking

In order for SFA to correctly identify the expected PMI, the STEP files must strictly follow the following naming convention:

```
nist-stc-09-systemcode-242.stp
```

For instance, `nist-stc-09-do-242.stp` would be the STEP file exported by Datakit out of Creo for the STC-09 model.

<u>*Note*</u> that even though this is the PDC test case, the files shall be named after the model (STC-09), so that SFA can correctly identify the expected PMI.

2.7.4.4 Testing Considerations

Preprocessor (CAD System)

The preprocessing CAD system will export the test model such that

- UUIDs shall be present on Product, Product Version, Geometry (advanced_face and hole axis only) and Topology (edge_curves and closed_shells), PMI, and UDAs as above.

- Note that cylindrical advanced_faces and circular edge_curves for systems that export holes (or bosses) as half-cylinders and half-edges will assign UUIDs to those paired entities as single UUID assignments for pairs of half-cylinders or for pairs of half-edges collected in uuid_set_items.
- Advanced Face entities for all Hole Features in the model are collected in Shape Aspects for each hole and the Shape Aspect for each hole “feature” will be assigned a UUID.
- For any Hole Patterns, Hole Feature Shape Aspects in each pattern in the model are collected in a Composite Shape Aspect and the Composite Shape Aspect for each hole pattern will be assigned a UUID.
- UUIDs are present on semantic text from global model notes PMI. See illustration for details.

Postprocessor (CAD System)

The postprocessing system checks for

- UUIDs on Product, Product Version, Geometry and Topology, PMI, and UDAs as above
- UUIDs on semantic text from global model notes PMI
- UUIDs for the Shape Aspects (generated by preprocessors on export) of each hole feature or other cutout feature (non-circular)
- UUIDs for the Composite Shape Aspects for each hole pattern

Postprocessor (Downstream System)

The postprocessing system will, based on the RP for CDE, read and process a mapping table for STEP entities and their UUIDs and check for

1. UUIDs on Product, Product Version, Geometry and Topology, PMI, and UDAs enti-ties as allowed by their model ingestion process.
2. Connect their internal features and characteristics to the STEP entities using the methods described in the RP for CDE

2.7.5 Statistics

For each STEP file exported or imported during one of the iterations of the PDC test case, vendors must submit the corresponding statistics. To do so, go to the [PDC Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e. test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results found after processing the file as described below.

Data Sheet Columns

column name	description
model	The name of the test model, here 'PDC'.
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system or downstream system importing the STEP file. For native stats, select 'stp'
pid_product	pass/fail - whether the persistent ID for the model at the product level was transferred correctly
pid_version	pass/fail – whether the persistent ID for the model version at the product level was transferred correctly
num_pid_pmi	The number of Persistent IDs assigned to Semantic PMI elements
num_pid_sfcs	The number of Persistent IDs assigned to surface elements (i.e., `advanced_face`)
num_pid_topol	The number of Persistent IDs assigned to topological elements (i.e., `edge_curves` only)
num_pid_shape	The number of Persistent IDs assigned to `shape_aspects`
num_pid_pattern	The number of Persistent IDs assigned to `composite_shape_aspects` for patterns of hole "features" processed.
num_pid_sgeom	The number of Persistent IDs assigned to Supplemental Geometry elements processed
num_pid_uda	The number of Persistent IDs assigned to User Defined Attribute elements processed
num_pid_sem_text	The number of Persistent IDs assigned to semantic text PMI elements processed
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.8 Test Case PDI: Persistent IDs for Design Iteration (Cases 0-4)

All information about this test case can also be viewed in CAESAR on its Information page.

2.8.1 Motivation

The ability to track a product's model information during design iteration has been limited by the lack of support for persistent IDs in STEP. With the inclusion of persistent IDs, collaborating systems are now able to exchange model data and track that data during design iteration.

In earlier test rounds, the design iteration test case focused on the stability of Persistent IDs (PIDs) between versions of the source design model as the model was modified and re-exported by the source system from iteration to iteration. Those initial tests showed that the PIDs could be exchanged and remain the same for model objects even when the location, size, or other properties of a model object were modified during an iteration. In R56J, the design iteration test case was extended to test bi-directional exchange of a model with persistent IDs where PIDs are assigned by the preprocessor at export and received by the postprocessor on import as reference PIDs. In a second exchange, the roles are reversed. The preprocessor (formerly the postprocessor) adds a new feature with a new PID and PMI on that feature with its own PMI and exports the model with the original source system's PIDs as reference and its own feature's PID as assigned. When the postprocessor (formerly the preprocessor) imports the model, it marks the received reference PIDs as its own assigned PIDs and captures the new geometric feature and its PMI with their new assigned PIDs and marks those PIDs as reference.

In this test round, both the original PDI "iteration cases" with iterations 0 through 4 will be tested (this test case, PDI) as well as the "Send and Receive" case (see test case, PDSR).

In this test case, the introduction and exchange of persistent IDs provides the ability of any contributor to the information stream associated with a product's lifecycle to add information to the model that can be connected to existing model content and that additional information can be retrieved by subsequent users and used as shared design content or feedback from the contributor.

Though separate test cases (PDC & PDU) focus on the downstream consumption of persistent IDs, such as manufacturing and inspection, the PD04 iteration data set will also be exchanged for downstream consumption.

2.8.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#391	PID Designer requirement
#392	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (initial use).
#413	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (first full process flow)
#415	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (revision capture/impact 1).
#417	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (revision capture/impact 2).

2.8.3 Approach

The approach to be used for publishing is described in the "Recommended Practices for Permanent Entity IDs for Design Iteration and Downstream Exchange" (Version 1.7; 29 October 2025), which can be found on the MBx-IF webpage under CAx > Rec. Practices.

The approach to be used for downstream consumption is described in the "Recommended Practices for Cross domain Exchange" (Draft Version 1.0; 17 December 2025), which can be found on the MBx-IF Nextcloud, folder

> **MBX-IF > CAX-IF > Draft Recommended Practices > Rec_Prac_for_Cross_Domain_Exchange.**

Within the domain of Persistent IDs, the following functionalities are in scope of Round 57J:

- Persistent IDs on Model (Product and Model Version) for
  - testing the retention of model ID information during exchange with downstream systems
- Persistent IDs on Geometry and Topology for
  - testing the retention of persistent IDs on geometry and topology with downstream applications that reference geometry and topology
  - This concept includes the introduction of Persistent IDs on shape_aspect, when needed, to collect individual geometry elements into logical groups such as

complex hole geometries and/or the collection of groups of shape aspects of patterned geometric groups.

    - When sending and receiving systems have differing topological structures, e.g. periodic or aperiodic cylindrical holes and edges, implementers will use the uuid_set_item for collecting these pairs and assign a single UUID to the pair.

- Persistent IDs on Semantic PMI Representation for
    - testing the assignment and exchange of persistent IDs on semantic PMI that references that geometry and topology for downstream consumption.
    - testing the assignment and exchange of persistent IDs on semantic PMI for dependent manufacturing planning that reference that semantic PMI.
    - testing the assignment and exchange of persistent IDs on semantic PMI on dependent metrology planning that reference that semantic PMI.
- Persistent IDs on Semantic Text (PMI) Representation for
    - testing the assignment and exchange of persistent IDs on semantic text used to represent global notes that likely will include “unless otherwise specified” requirements and their impact on dependent manufacturing planning that reference that semantic text.
    - testing the assignment and exchange of persistent IDs on semantic text used to represent global notes that likely will include “unless otherwise specified” requirements and their impact on dependent metrology planning that reference that semantic text.
- Persistent IDs on User Defined Attributes for
    - testing the assignment and exchange of persistent IDs on attributes and their values that may be used as reference data for dependent manufacturing planning.

The following are out of scope for Round 57J and are moved to the Future Considerations section:

- Persistent IDs on Geometry and Persistent IDs on Semantic PMI Representation for testing assembly constraints referencing those geometries.

In context of the “send and receive” use case for persistent IDs, the following functionalities are in scope for Round 57J:

- Persistent IDs on Model (Product and Model Version),
- Persistent IDs on Geometry and Topology,
- Persistent IDs on Semantic PMI Representation, and
- Persistent IDs on User Defined Attributes
- The key requirements for this test case are
    - to assign PIDs on data that is owned (created),
    - to capture as reference, PIDs on data that is received from others, and
    - to be able to display to the user, on command, all assigned PIDs and all reference PIDs and the entities to which they belong. [the method of display – model tree, parameter table, analysis table, or exported file and the content of the entity information displayed – is left to the vendor to determine].

The following are out of scope for Round 57J and are moved to the Future Considerations section:

- Persistent IDs on Geometry and Persistent IDs on Semantic PMI Representation for testing assembly constraints referencing those geometries.

To support the concepts in scope of this test case, the final AP242 Edition 4 schema shall be used, which can be found via the public MBx-IF homepage under “MBx > Resources > EXPRESS Schemas.”

2.8.4 Testing Instructions

The test will be performed based on a simple test model, developed by Rosemary Astheimer. The test case is a series of multiple exchanges in the same direction ("ping-ping") between two exchanging systems, with the sending system making changes to existing native model features or adding new model features before exchanging with their exchange partner system, testing the effect of model change during iterative design exchange.

2.8.4.1 Test Model Access

Native CAD files are available in CATIA V5-6R2023, NX 2207, and Creo 11.04 formats for the test case can be downloaded from Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 57J > PDI

2.8.4.2 Test Model Configuration

Similar to Round 55J, the PD04 test case requires iteration to confirm retention of persistent entity IDs. This test case is a multi-model, unidirectional iterative exchange process in which five exchanges will take place – an initial exchange and four subsequent exchanges in the same direction after different model design changes are made by the original sender.

The four iterative changes are described in the following sections and will be identified in CAESAR by the model suffixes given below:

- PDI_0-0 – Initial Exchange (axis of hole included)
- PDI_0-1 – First Updated Exchange (hole moved, axis of hole moved)
- PDI_0-2 – Second Updated Exchange (hole and axis deleted; PMI for hole deleted)
- PID_0-3 – Third Updated Exchange (hole recreated at same location as PDI_0-1 with axis and PMI)
- PID_0-4 – Fourth Updated Exchange (chamfer added)

Each iteration is intended to test the behavior of UUIDs during model exchange with changes resulting either movement of existing entities with their UUIDs or in new entities (and thus new UUIDs) added and entities removed (with their UUIDs discarded).

The models also include the introduction of a model level UDA which should have a UUID assigned.

For details on the test model configuration, refer to the presentation

CAx-IF_R56J_PID_Test_Cases_v3.pptx

provided in the Nextcloud folder linked above.

Note also that there are two mechanisms for supporting persistent IDs in STEP:

- The first is via the creation of new persistent uuid_attribute entities attached to certain entities within the Data Section of the Part 21 file.
- The second is via the creation of persistent ID relationships between STEP entity IDs and persistent entity IDs within an ANCHOR Section of a Part 21 Edition 3 file.

Please refer to the Recommended Practice document for further details about the valid entity subtypes to be used in place of the abstract uuid_attribute entity type in the Data Section. Based on agreement, the scope of test rounds (Round 54J and later) will include only the first type of ID, i.e. the v5_uuid_attribute subtype in the Data Section. This is reflected in the current version of the Recommended Practices. The testing of the ANCHOR Section approach will be considered in a future test round.

2.8.4.3 Test Case Iterations

The test case for design iteration is a sequence of two simple exchanges that represent an exchange between two designers, designated A and B respectively, or alternatively between a designer and a downstream consumer. This sequence of exchanges, each considered a sub-case of the PDI test case, is described below.

PDI_0 (initial exchange)

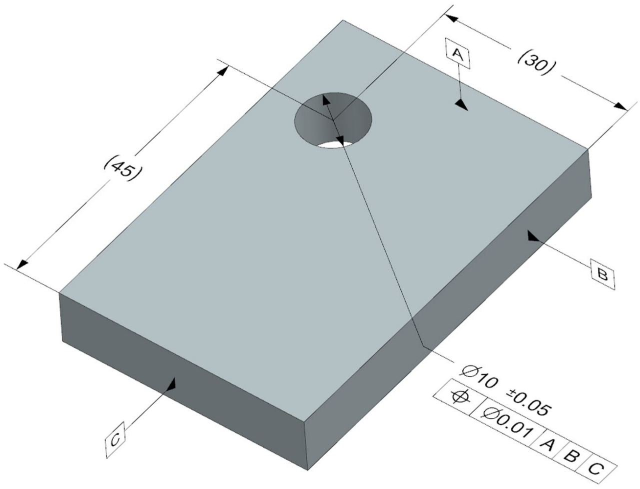


Figure *10: Illustration of PDI_0-0 Test Case*

Preprocessor (CAD System) A's initial design is exchanged to B (model version is 0-0).

Postprocessor (CAD System or Downstream System) B checks for

1. UUID on Product.
2. UUID on Product Version
3. UUIDs on all Advanced_Faces (either 7 or 8 depending on system)
4. UUIDs on all topological Edges (either 14 or 18 depending on system)
5. UUID on Closed Shell
6. UUID on MSBR
7. UUIDs on PMI (diameter and location tolerance, datum features, and hole location dimensions)
8. UUID on Axis through the hole (supplemental geometry)
9. UUIDs on UDAs (at least one exists in the native model, at the model level)

PDI_0-1 (First change - hole moves)

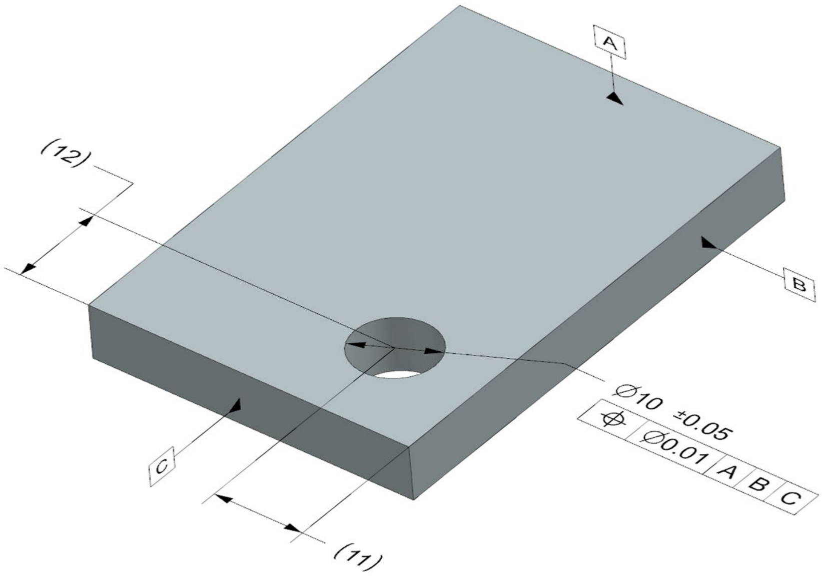


Figure 11: Illustration of PDI_0-1 Test Case

Preprocessor (CAD System) A modifies the location of the hole (moved, not replaced), versions model, and resends revised model to B

Postprocessor (CAD or Downstream System) B checks for

1. Product Version has changed and UUID on Product Version is the same
2. Model hole surface (or surfaces) move and PMI dimensions remain associated with geometry
3. All above UUIDS (product, advanced_faces, edge_curves, closed_shell, MSBR, axis, PMI, and UDA(s)) should be the same as previously imported
    a. 1 Product
    b. 1 Product Version (Version identifier should change)
    c. 7 or 8 Faces
    d. 14 or 18 Edges
    e. 1 Closed Shell
    f. 1 MSBR
    g. 7 PMI (Diameter and Location Tolerance, datums, and hole location dimensions)
    h. Axis through the hole (supplemental geometry)
    i. UDA(s) – at least one at the model level

PDI_0-2 (Second change - hole, hole axis, and hole PMI deleted)

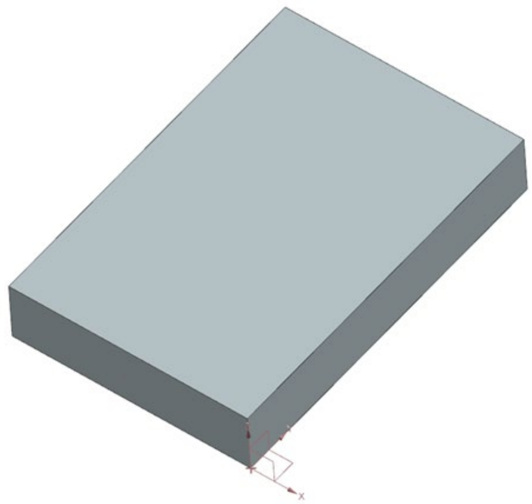


Figure 12: Illustration of PDI_0-2 Test Case

Preprocessor (CAD System) In case 2, A deletes the hole, the axis, and the PMI associated with the hole, versions the model, and resends revised model to B

Postprocessor (CAD System) B checks for

1. Product UUID is the same.
2. UUID on Product Version is the same; Product Version value has changed.
3. Since the hole, the axis, and the PMI associated have been deleted, the UUIDs for the hole (Faces and Topology), the axis and the PMI associated with the hole are marked as removed.
4. PMI UUIDs (datums) are the same. PMI UUIDs on Hole Diameter and Location Tolerance and Hole Dimensions, which are replaced when the hole is replaced [if exchanged, optional].
5. UUIDs on UDAs (at least one at the model level, and it will be the same).

PDI_0-3 (Third change - hole, hole axis, and hole PMI recreated)

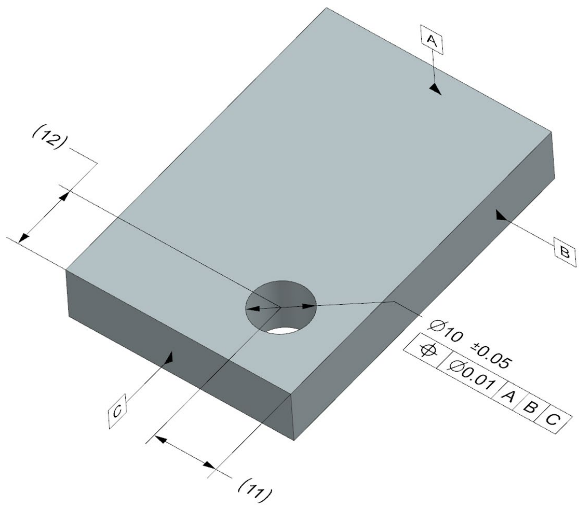


Figure 13: Illustration of PDI_0-3 Test Case

Preprocessor (CAD System) In case 3, A recreates the hole, hole axis, and hole PMI, versions model, and resends revised model to B

Postprocessor (CAD System) B checks for

1. Product UUID is the same.
2. UUID on Product Version is the same; Product Version value has changed.
3. Original UUIDs for the block's Datum Features are the same.
4. Since the hole and its axis have been recreated, the UUIDs for the hole (Faces and Topology), and for the axis are changed.
5. PMI UUIDs (datum features) are the same, however for PMI UUIDs on Hole Diameter and Location Tolerance and Hole Dimensions, which were recreated, will be new.
6. UUIDs on UDAs (at least one at the model level, and it will be the same).

PDI_0-4 (Fourth change - chamfer added)

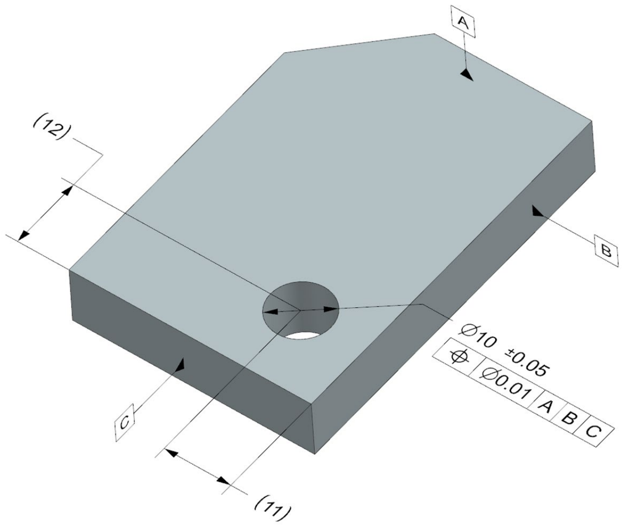


Figure 14: Illustration of PDI_0-4 Test Case

Preprocessor (CAD System) A adds a chamfer to the corner of the block, versions model, and sends model back to B

Postprocessor (CAD System or Downstream System) B checks for

1. Product UUID is the same.
2. UUID on Product Version is the same; Product Version value has changed.
3. Original UUIDs for the block are the same (except for the changes noted in 5 below).
4. PMI UUIDs are the same [if exchanged, optional].
5. New UUIDs (generated by A on export) for the new chamfer face and its new topological edges.
6. UUIDs on UDAs (at least one at the model level, and it will be the same).

2.8.5 Statistics

For each STEP file exported or imported during one of the iterations of the PDI test case, vendors must submit the corresponding statistics. To do so, go to the [PDI Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e. test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results found after processing the file as described below.

Data Sheet Columns

column name	description
model	The name of the test model, here 'PDI'. **Important:** Add the iteration as suffix to the model, i.e.: • PDI_0 for the initial exchange • PDI_1 for the first design change (hole move) • PDI_2 for the first alternate design change (hole replace) • PDI_3 for the second design change (chamfer add)
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system or downstream system importing the STEP file. For native stats, select 'stp'
pid_product	pass/fail - whether the persistent ID for the model at the product level was transferred correctly
pid_version	pass/fail – whether the persistent ID for the model version at the product level was transferred correctly
num_pid_pmi	The number of Persistent IDs assigned to Semantic PMI elements
num_pid_sfcs	The number of Persistent IDs assigned to surface elements (i.e., `advanced_face`)
num_pid_topol	The number of Persistent IDs assigned to topological elements (i.e., `edge_curves` only)
num_pid_sgeom	The number of Persistent IDs assigned to Supplemental Geometry elements
num_pid_uda	The number of Persistent IDs assigned to User Defined Attribute elements
num_pid_sem_text	The number of Persistent IDs assigned to semantic text PMI elements
pid_ownership	all/partial/none – Indicates whether the receiving system was able to correctly manage the ownership for Persistent IDs (refer to PID Rec. Pracs. for details)
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.9 Test Case PDSR: Persistent IDs for Design Iteration (Send & Receive)

All information about this test case can also be viewed in CAESAR on its Information page.

2.9.1 Motivation

The ability to track a product's model information during design iteration has been limited by the lack of support for persistent IDs in STEP. With the inclusion of persistent IDs, collaborating systems are now able to exchange model data and track that data during design iteration.

In earlier test rounds, the design iteration test case focused on the stability of Persistent IDs (PIDs) between versions of the source design model as the model was modified and re-exported by the source system from iteration to iteration. Those initial tests showed that the PIDs could be exchanged and remain the same for model objects even when the location, size, or other properties of a model object were modified during an iteration. In R56J, the design iteration test case was extended to test bi-directional exchange of a model with persistent IDs where PIDs are assigned by the preprocessor at export and received by the postprocessor on import as reference PIDs. In a second exchange, the roles are reversed. The preprocessor (formerly the postprocessor) adds a new feature with a new PID and PMI on that feature with its own PMI and exports the model with the original source system's PIDs as reference and its own feature's PID as assigned. When the postprocessor (formerly the preprocessor) imports the model, it marks the received reference PIDs as its own assigned PIDs and captures the new geometric feature and its PMI with their new assigned PIDs and marks those PIDs as reference.

In this test round, both the original PDI "iteration cases" with iterations 0 through 4 will be tested (see test case PDI) as well as the "Send and Receive" case (this test case, PDSR).

In this test case, the introduction and exchange of persistent IDs provides the ability of any contributor to the information stream associated with a product's lifecycle to add information to the model that can be connected to existing model content and that additional information can be retrieved by subsequent users and used as shared design content or feedback from the contributor.

2.9.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#391	PID Designer requirement
#392	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (initial use).
#413	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (first full process flow)
#415	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (revision capture/impact 1).
#417	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (revision capture/impact 2).

2.9.3 Approach

The approach to be used for publishing is described in the "Recommended Practices for Permanent Entity IDs for Design Iteration and Downstream Exchange" (Version 1.7; 29 October 2025), which can be found on the MBx-IF webpage under CAx > Rec. Practices.

Within the domain of Persistent IDs, the following functionalities are in scope of Round 57J:

- Persistent IDs on Model (Product and Model Version) for
  - testing the retention of model ID information during exchange with downstream systems
- Persistent IDs on Geometry and Topology for
  - testing the retention of persistent IDs on geometry and topology with downstream applications that reference geometry and topology
  - This concept includes the introduction of Persistent IDs on shape_aspect, when needed, to collect individual geometry elements into logical groups such as complex hole geometries and/or the collection of groups of shape aspects of patterned geometric groups.
  - When sending and receiving systems have differing topological structures, e.g. periodic or aperiodic cylindrical holes and edges, implementers will use the uuid_set_item for collecting these pairs and assign a single UUID to the pair.
- Persistent IDs on Semantic PMI Representation for
  - testing the assignment and exchange of persistent IDs on semantic PMI that references that geometry and topology for downstream consumption.
  - testing the assignment and exchange of persistent IDs on semantic PMI for dependent manufacturing planning that reference that semantic PMI.
  - testing the assignment and exchange of persistent IDs on semantic PMI on dependent metrology planning that reference that semantic PMI.
- Persistent IDs on Semantic Text (PMI) Representation for
  - testing the assignment and exchange of persistent IDs on semantic text used to represent global notes that likely will include “unless otherwise specified” requirements and their impact on dependent manufacturing planning that reference that semantic text.
  - testing the assignment and exchange of persistent IDs on semantic text used to represent global notes that likely will include “unless otherwise specified” requirements and their impact on dependent metrology planning that reference that semantic text.
- Persistent IDs on User Defined Attributes for
  - testing the assignment and exchange of persistent IDs on attributes and their values that may be used as reference data for dependent manufacturing planning.

The following are out of scope for Round 57J and are moved to the Future Considerations section:

- Persistent IDs on Geometry and Persistent IDs on Semantic PMI Representation for testing assembly constraints referencing those geometries.

In context of the “send and receive” use case for persistent IDs, the following functionalities are in scope for Round 57J:

- Persistent IDs on Model (Product and Model Version),
- Persistent IDs on Geometry and Topology,
- Persistent IDs on Semantic PMI Representation, and
- Persistent IDs on User Defined Attributes

- The key requirements for this test case are
  - to assign PIDs on data that is owned (created),
  - to capture as reference, PIDs on data that is received from others, and
  - to be able to display to the user, on command, all assigned PIDs and all reference PIDs and the entities to which they belong. [the method of display – model tree, parameter table, analysis table, or exported file and the content of the entity information displayed – is left to the vendor to determine].

The following are out of scope for Round 57J and might be considered in the future:

- Persistent IDs on Geometry and Persistent IDs on Semantic PMI Representation for testing assembly constraints referencing those geometries.

To support the concepts in scope of this test case, the final AP242 Edition 4 schema shall be used, which can be found via the public MBx-IF homepage under “MBx > Resources > EXPRESS Schemas.”

2.9.4 Testing Instructions

The test will be performed based on a simple test model, developed by Rosemary Astheimer. The test case is a pair of exchanges going back and forth between two exchanging systems ("ping-pong"), with the first (sending) system creating the model and the second (receiving) system making changes to the imported model features, or adding new model features, before returning it to their exchange partner, testing the effect of model change during iterative design exchange.

2.9.4.1 Test Model Access

Native CAD files are available in CATIA V5-6R2023, NX 2207, and Creo 11.04 formats for the test case can be downloaded from Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 57J > PDI

2.9.4.2 Test Model Configuration

The PDI test case in this test round requires managing the ownership of persistent entity IDs as the models are exchanged. This test case is a single model, bi-directional exchange process in which two exchanges will take place – an initial exchange in one direction and a second subsequent exchange in the opposite direction after one model design change (feature and PMI on that feature) are made by the sender in the return exchange (formerly the receiver in the first exchange).

The two iterative changes are described in the following sections and will be identified in CAESAR by the model suffixes given below:

- PDSR_S – Initial Exchange (Sent)
- PDSR_R – Updated Exchange (Returned, with a chamfer added to the hole with PMI for the chamfer dimension also added)

Note:

- The native model to be used for "PDSR_S" is the initial iteration step "0-0" from the packages linked above.
- There is no native model for "PDSR_R", as this is the result of importing a "PDI_S" STEP file and then applying the modifications described below.

For details on the test model configuration, refer to the presentation

CAx-IF_R56J_PID_Test_Cases_v3.pptx

provided in the Nextcloud folder linked above.

Each iteration is intended to test the management of UUID ownership (see PID Rec. Practices for details).

Note also that there are two mechanisms for supporting persistent IDs in STEP:

- The first is via the creation of new persistent uuid_attribute entities attached to certain entities within the Data Section of the Part 21 file.
- The second is via the creation of persistent ID relationships between STEP entity IDs and persistent entity IDs within an ANCHOR Section of a Part 21 Edition 3 file.

Please refer to the Recommended Practice document for further details about the valid entity subtypes to be used in place of the abstract uuid_attribute entity type in the Data Section. Based on agreement, the scope of test rounds (Round 54J and later) will include only the first type of ID, i.e. the v5_uuid_attribute subtype in the Data Section. This is reflected in the current version of the Recommended Practices. The testing of the ANCHOR Section approach will be considered in a future test round.

2.9.4.3 Test Case Iterations

The test case for design iteration is a sequence of two simple exchanges that represent an exchange between two designers, designated A and B respectively, or alternatively between a designer and a downstream consumer. This sequence of exchanges, each considered a sub-case of the PDI test case, are described and illustrated below.

PDSR_S (initial exchange)

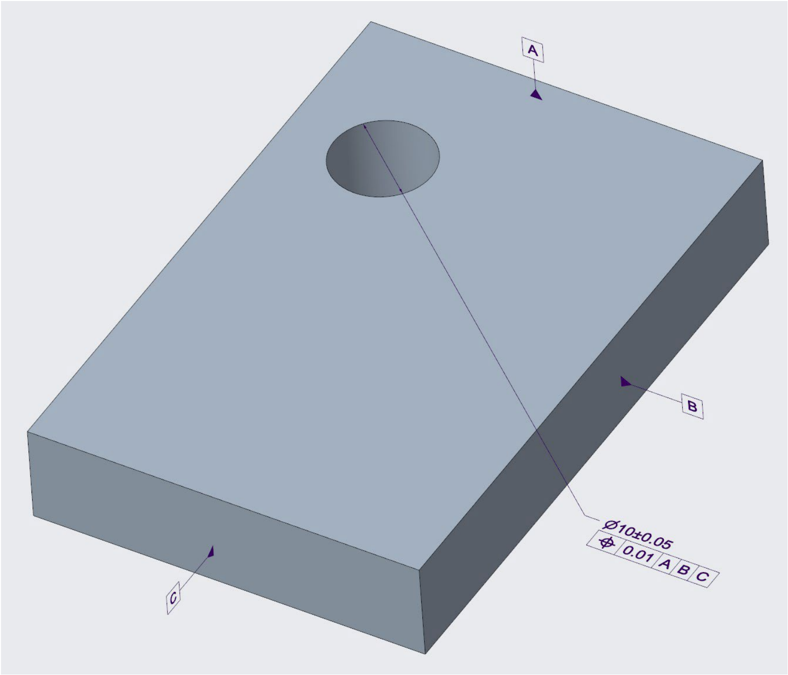


Figure *15: Illustration of PDI_S Test Case*

Preprocessor (CAD System) A's initial design is exchanged to B (model version is S).

Postprocessor (CAD System) B checks for

1. UUID on Product (marked as Reference)
2. UUID on Product Version (marked as Reference)
3. UUIDs on all Advanced_Faces (either 7 or 8 depending on system); all marked as Reference

4. UUIDs on all topological Edges (either 14 or 18 depending on system); all marked as Reference
5. UUID on Closed Shell; marked as Reference
6. UUID on MSBR; marked as Reference
7. UUIDs on PMI (diameter and location tolerance, datum features, and hole location di-men-sions); marked as Reference
8. UUID on Axis through the hole (supplemental geometry); marked as Reference
9. UUIDs on UDAs (at least one exists in the native model, at the model level); marked as Reference

PDSR_R (First change – a chamfer and PMI on that chamfer are added)

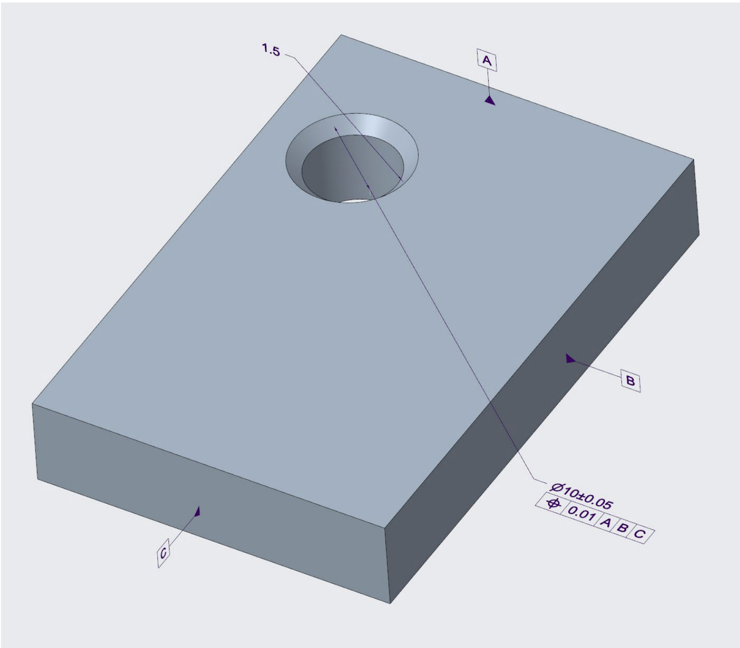


Figure 16: Illustration of PDI_R Test Case

Preprocessor (CAD System) B modifies the hole by adding the chamfer and its PMI, versions model, and returns the revised model to A (the UUIDs for the chamfer surfaces, edges, and chamfer PMI are marked as assigned, all other UUIDs are marked as reference)

Postprocessor (CAD System) A checks for

- Product is the same and UUID on Product is the same; UUID is marked as assigned
- Product Version has changed and UUID on Product Version is the same; UUID is marked as assigned
- Model hole has a new chamfer and PMI dimension associated to the chamfer geometry; UUID is marked as reference

All above UUIDS (product, advanced_faces, edge_curves, closed_shell, axis, PMI, and UDA(s)) should be the same as previously exported, i.e., assigned, except the new chamfer faces, edges, and chamfer PMI are marked as reference.

2.9.5 Statistics

For each STEP file exported or imported during one of the iterations of the PDSR test case, vendors must submit the corresponding statistics. To do so, go to the [PDSR Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e. test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Target Statistics

When importing a STEP file, report the results found after processing the file as described below.

Data Sheet Columns

column name	description
model	The name of the test model, here 'PDI'. **Important:** Add the iteration as suffix to the model, i.e.: • PDI_S for the initial exchange • PDI_R for the returned model with the design change (added chamfer with PMI)
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system or downstream system importing the STEP file. For native stats, select 'stp'
pid_product	pass/fail - whether the persistent ID for the model at the product level was transferred correctly
pid_version	pass/fail – whether the persistent ID for the model version at the product level was transferred correctly
num_pid_pmi	The number of Persistent IDs assigned to Semantic PMI elements
num_pid_sfcs	The number of Persistent IDs assigned to surface elements (i.e., `advanced_face`)
num_pid_topol	The number of Persistent IDs assigned to topological elements (i.e., `edge_curves` only)
num_pid_sgeom	The number of Persistent IDs assigned to Supplemental Geometry elements
num_pid_uda	The number of Persistent IDs assigned to User Defined Attribute elements
num_pid_sem_text	The number of Persistent IDs assigned to semantic text PMI elements
pid_ownership	all/partial/none – Indicates whether the receiving system was able to correctly manage the ownership for Persistent IDs (refer to PID Rec. Pracs. for details)
date	The date when the statistics were last updated (will be filled in automatically)
issues	A short statement on issues with the file

2.10 Test Case PDU: Persistent IDs for Downstream Consumption with Surface Texture and Unless Otherwise Specified (General) Requirements

All information about this test case can also be viewed in CAESAR on its Information page.

2.10.1 Motivation

The ability to track a product's model information during design iteration has been limited by the lack of support for persistent IDs in STEP. With the inclusion of persistent IDs, collaborating systems are now able to exchange model data and track that data during downstream consumption.

In this test round we introduce a new capability - Surface Texture, available in the STEP standard for some time but, to date, never having had a Recommended Practices document to support implementation by the vendors. Support for Surface Texture is a significant requirement for downstream manufacturing and quality processes. As part of this new test case, we will also explore the annotation of a model with PMI representing both local requirements and general, model level requirements. These general, model level requirements are often modeled as model level notes starting with the phrase "Unless Otherwise Specified". We will exercise these concepts for surface texture and other characteristics important to downstream use in this test case and will include Persistent ID assignment for both local and global requirements.

2.10.2 User Stories

This test case supports the following User Stories provided by the CAx-IF UG on Redmine:

ID	Title
#189	Surface roughness
#391	PID Designer requirement
#392	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (initial use).
#413	PID CAI- traceable data element relationship through multiple application ingestion and formatting change (first full process flow)

2.10.3 Approach

The approach to be used for publishing is described in the "Recommended Practices for Permanent Entity IDs for Design Iteration and Downstream Exchange" (Version 1.7; 29 October 2025), which can be found on the MBx-IF webpage under CAx > Rec. Practices.

The approach to be used for downstream consumption is described in the "Recommended Practices for Cross domain Exchange" (Draft Version 1.0; 17 December 2025), which can be found on the MBx-IF Nextcloud, folder

MBX-IF > CAX-IF > Draft Recommended Practices > Rec_Prac_for_Cross_Domain_Exchange.

The approach to be used for Surface Texture is described in the "Recommended Practices for Surface Texture" (Version 0.2; 15 January 2026), which can be found on Nextcloud, folder

MBX-IF > CAX-IF > Draft Recommended Practices

In context of the "downstream consumption" use case for persistent IDs, the following functionalities are in scope for Round 57J:

- Persistent IDs on all entities previously described under other PID test cases
- Persistent IDs on Geometry and Topology,
- Persistent IDs on Semantic PMI Representation, and

- Persistent IDs on User Defined Attributes
- The key requirements for this test case are
  - to assign and exchange PIDs on surface texture and geometric tolerance data that is attached to local geometry,
  - to assign and exchange PIDs on surface texture and geometric tolerance data that is captured in global "unless otherwise specified" notes, and
  - to be able to display to the user, on command, all assigned PIDs or all reference PIDs and the entities to which they belong. [the method of display – model tree, parameter table, analysis table, or exported file and the content of the entity information displayed – is left to the vendor to determine].

The following are out of scope for Round 57J and are moved to the Future Considerations section:

- Persistent IDs on Geometry and Persistent IDs on Semantic PMI Representation for testing assembly constraints referencing those geometries.

To support the concepts in scope of this test case, the final AP242 Edition 4 schema shall be used, which can be found via the public MBx-IF homepage under "MBx > Resources > EX-PRESS Schemas."

2.10.4 Testing Instructions

The test will be performed using a variant of the now well-known PDI_0 model, renamed PDU (see below). The test case is a single exchange between two systems. No model changes will be made.

2.10.4.1 Test Model Access

A new dedicated native model has been created for the PDU test case. It is currently available only in Creo 12.4.0 format and can be downloaded from Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 57J > PDU

Native models in other CAD formats (CATIA V5, NX) will be added at a later point in time.

2.10.4.2 Test Model Configuration

The PDU test case in this test round requires exchanging persistent entity IDs on both local and global instances of surface texture and geometric tolerances in the model exchanged. This test case is a single model, unidirectional exchange.

Implementors should refer to the document "*pdu_R56J_FINAL.pdf*" in the *R56J > PDU* test case folder to help identify all PMI in the models and their associated reference geometry elements for the PDU_R56J_FINAL test case.

For details on the test model configuration, refer to the presentation

CAx-IF_R56J_PID_Test_Cases_v3.pptx

provided in Nextcloud, folder

MBX-IF > CAX-IF > CAX-IG > Round 56J > PDI

Note also that there are two mechanisms for supporting persistent IDs in STEP:

- The first is via the creation of new persistent uuid_attribute entities attached to certain entities within the Data Section of the Part 21 file.
- The second is via the creation of persistent ID relationships between STEP entity IDs and persistent entity IDs within an ANCHOR Section of a Part 21 Edition 3 file.

Please refer to the Recommended Practice document for further details about the valid entity subtypes to be used in place of the abstract uuid_attribute entity type in the Data Section. Based on agreement, the scope of test rounds (Round 54J and later) will include only the first type of ID, i.e. the v5_uuid_attribute subtype in the Data Section. This is reflected in the current version of the Recommended Practices. The testing of the ANCHOR Section approach will be considered in a future test round.

PDU (single exchange)

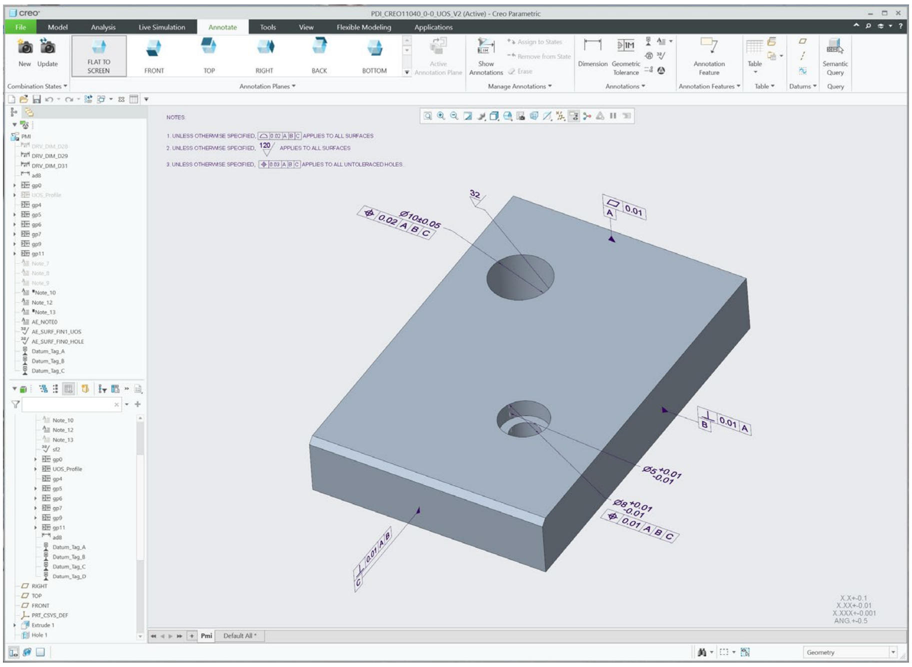


Figure *17: Illustration of PDU Test Case*

Preprocessor (CAD System)

- the preprocessing system will assign PIDs to all data as in other PID test cases and, in particular, assignment of PIDs to both local and global surface finish and geometric tolerances.

Postprocessor (CAD System) B checks for

1. UUID on Product
2. UUID on Product Version
3. UUIDs on all Advanced_Faces (either 7 or 8 depending on system)
4. UUIDs on all topological Edges (either 14 or 18 depending on system)
5. UUID on Closed Shell
6. UUID on MSBR
7. UUIDs on PMI (diameter and location tolerance, datum features, and hole location dimensions)
8. UUID on Axis through the hole (supplemental geometry)
9. UUIDs on UDAs (at least one exists in the native model, at the model level)
10. UUIDs on local surface finish and local profile tolerances
11. UUIDs on global surface finish, global profile tolerance, and global location tolerance

Postprocessor (Downstream Systems)

The postprocessing system will, based on the RP for CDE, read and process a mapping table for STEP entities and their UUIDs and check for

1. UUIDs on Product, Product Version, Geometry and Topology, PMI, UDAs, and Surface Texture entities as allowed by their model ingestion process.
2. UUIDs on Unless Otherwise Specified global requirements, i.e. surface texture, pro-file tolerance, and true position tolerance
3. Connect their internal features and characteristics to the STEP entities using the methods described in the RP for CDE.

2.10.5 Statistics

For each STEP file exported or imported during the exchange of the R57J PDU test case, vendors must submit the corresponding statistics. To do so, go to the [PDU Data Sheet], and either fill in the web form, or upload a comma-delimited file (.csv) with the data as listed below.

Native Statistics

When exporting a STEP file, report what data importing systems should expect to find. For numeric statistics, enter the respective value or 'na' if not supported. For other statistics, select either 'full support' (i.e. test case and Rec. Pracs. definitions are fulfilled), 'limited support' (meaning the implementation does not meet all criteria and issues may be expected on import), or 'na' if not supported.

Please include a snapshot image of the list of assigned or reference PIDs in the model and what they point to.

Target Statistics

When importing a STEP file, report the results found after processing the file as described below.

Please include a snapshot image of the list of assigned or reference PIDs in the model and what they point to.

Data Sheet Columns

column name	description
model	The name of the test model, here 'PDU'.
system_n	The system code of the CAD system creating the STEP file
system_t	The system code of the CAD system importing the STEP file. For native stats, select 'stp'
pid_product	pass/fail – whether the persistent ID at the product level was transferred correctly
pid_version	pass/fail – whether the model version at the product level was transferred correctly
num_pid_pmi	The number of Persistent IDs assigned to Semantic PMI elements
num_pid_sfcs	The number of Persistent IDs assigned to surface elements (i.e., `advanced_face`)
num_pid_topol	The number of Persistent IDs assigned to topological elements (i.e., `edge_curves` only)
num_pid_sgeom	The number of Persistent IDs assigned to Supplemental Geometry elements

column name	description
num_pid_uda	The number of Persistent IDs assigned to User Defined Attribute elements
surface_textures	The number of surface texture elements defined in the file.
num_pid_surf_txt	The number of Persistent IDs on Surface Texture elements
surface_tex-ture_params	The number of surface texture parameters defined in the file.
num_pid_surf_param	The number of Persistent IDs on Surface Texture Parameters
date	The date when the statistics were last updated (will be filled in au-tomatically)
issues	A short statement on issues with the file